UNIVERSIDADE LUSÍADA DE VILA NOVA DE FAMALICÃO

DESENVOLVIMENTO DE UMA FERRAMENTA AUTOMATIZADA PARA CONSOLIDAR INFORMAÇÃO SOBRE FREGUESIAS PORTUGUESAS

Fábio André Lopes de Oliveira

Dissertação para obtenção do Grau de Mestre

Vila Nova de Famalicão 2013

UNIVERSIDADE LUSÍADA DE VILA NOVA DE FAMALICÃO

DESENVOLVIMENTO DE UMA FERRAMENTA AUTOMATIZADA PARA CONSOLIDAR INFORMAÇÃO SOBRE FREGUESIAS PORTUGUESAS

Fábio André Lopes de Oliveira

Dissertação para obtenção do Grau de Mestre

Orientador: Prof. Dr. Vítor Pereira

Vila Nova de Famalicão 2013

Agradecimentos

Tenho primeiramente que agradecer à Universidade Lusíada de Vila Nova de Famalicão pela oportunidade que me deu de realizar um projecto desta dimensão, ao Orientador Vítor Pereira pela paciência e pela sua grande ajuda na elaboração deste projecto, aos meus familiares e amigos que são uma grande fonte de apoio, onde sou sempre recebido com carinho, palavras de confiança e de alento, e também quero gratificar todos aqueles que de forma directa ou indirecta contribuíram para a execução desta dissertação.

Índice

Resumo

A colossal expansão da internet é responsável por uma criação substancial de conhecimento que deve ser guardado e mantido em bases de dados centralizadas que por sua vez também aumentam em tamanho e complexidade. Manter e actualizar as informações armazenadas nas bases de dados não pode ser feito de forma eficiente pelos seres humanos por si só; ferramentas automatizadas têm sido usadas há algum tempo, com vários graus de sucesso. Uma das primeiras ferramentas de *software* a surgir foi o "*web crawler*", que é a base de como os motores de busca trabalham. Outra classe importante de ferramentas, chamada "*internet bots*", ou simplesmente "*bots*" (da palavra "robot"), é usado para ajudar os humanos a gerir grandes quantidades de dados.

Apesar da Wikipédia usar *bots* há mais de 10 anos, as páginas *web* de freguesias portuguesas estão frequentemente desactualizadas ou têm informações insuficientes. Além disso, os dados que podem ser utilizados para actualizar estas páginas web estão espalhados por várias fontes e têm um formato que não permite uma fácil comparação entre duas ou mais freguesias.

Este trabalho descreve o desenvolvimento de uma ferramenta automatizada para reunir informações de várias fontes (*online* e *offline*) sobre freguesias portuguesas, seguindo os passos principais da Engenharia de Software, isto é, especificação de requisitos, desenho, implementação e testes.

O resultado deste trabalho específico para um caso em particular demonstra a construção de uma ferramenta simples e acessível tanto a utilizadores básicos como a avançados, de maneira a usá-la para extrair informações sobre freguesias portuguesas.

Abstract

The remarkable growth of the Internet accounts for a substantial creation of knowledge. This knowledge is stored in centralized databases, which have increased in size and complexity. Maintaining and updating the information stored in databases cannot be done efficiently by humans alone; automated tools have been used for quite some time with various degrees of success. One of the first software tools to emerge was the "web crawler", which is the basis of how search engines work. Another important class of tools, called "internet bots" or simply “bots” (from the word “robot”), is used to help humans manage large quantities of data.

Even though Wikipedia uses bots for over 10 years, the web pages of Portuguese civil parishes are frequently outdated or have insufficient information. In addition, the data that can be used to update these web pages is scattered in various sources and in a format that does not allow an easy comparison between two or more parishes. For instance, an organization may need to compare the distribution of population from various parishes according to the number of people per family, age group or marital status.

This work describes the development of an automated tool to gather information from various sources (both online and offline) about Portuguese civil parishes (“*freguesias*” in Portuguese), following the main steps of Software Engineering namely, requirement specification, application design and implementation, and testing.

The result of this particular work for a particular case demonstrates the construction of an easy and accessible tool that both basic and advanced users can use to extract information about Portuguese civil parishes.

Palavras-Chave

Internet Bots

Web crawlers

HTML *Parsing*

Wikipédia

Freguesias

Actualização

Engenharia de Software

Bases de Dados

Informação

Lista de Abreviaturas

ANAFRE – Associação Nacional de Freguesias

AR – Assembleia da República

ATA – Autoridade Tributária e Aduaneira

CAOP – Carta Administrativa Oficial de Portugal

CAPTCHA – *Completely Automated Public Turing test to tell Computers and Humans Apart*

CSV – *Comma-Separated Values*

DoS – *Denial of Service*

HTML – *Hypertext Markup Language*

IGESPAR – Instituto de Gestão do Património Arquitectónico e Arqueológico

IGP – Instituto Geográfico Português

INE – Instituto Nacional de Estatística

MMORPG – *Massive Multiplayer Online Role Playing Game*

NPC – *Non-Player Character*

OLEDB – Object Linking and Embedding Database

WWW – *World Wide Web*

XLS – *Excel Binary File Format*

1. Introdução

O incrível crescimento em dimensão da internet é responsável pela criação de cada vez mais conhecimento, um importante activo capaz de impulsionar empresas e estimular nações. Contudo, com este aumento também cada vez mais toda essa informação fica espalhada por diversos *websites* e por vezes dispersa por vários ficheiros dentro do mesmo. A consequência destes dados estarem distribuídos é a dificuldade de os recolher e organizar, por exemplo, para actualizar uma base de dados, por isso é um grande desafio para um utilizador comum recolher toda essa informação.

Como Portugal atravessa uma crise económica, por vezes estas realidades são postas em segundo plano. Devido a uma recente lei (AR, 2012) irá existir uma reorganização do território para um incremento da eficiência e uma estimulação do desenvolvimento das administrações locais. O governo português prepara uma redução do número das freguesias na ordem das mil (Silva, 2013), o que representa cerca de 25% das 4259 freguesias existentes (IGP, 2011). Uma freguesia é uma administração secundária local, logo abaixo do concelho, que representa uma mais prática interacção, acessibilidade a informação da região ou informar qualquer tipo de irregularidade, disponibilizando para os habitantes de uma determinada localidade uma administração que ajuda a resolver estas questões. Com a agregação das freguesias, grandes quantidades de informação vão ficar desactualizadas, por vezes mesmo inconsistentes e obsoletas, como por exemplo na situação da combinação de várias freguesias numa só e adoptando um dos nomes existentes.

Apesar de um censos ter sido realizado recentemente (INE, 2012), parte dos dados deixam de ser verdadeiramente precisos com o desaparecimento de várias freguesias. Portanto é bastante trabalhoso o acto de fazer comparações entre freguesias, visto que os dados se encontram dispersos por várias plataformas (tanto *online* como *offline*) que por sua vez podem não estar actualizadas.

A manutenção e actualização destes dados não pode de modo algum ser realizada eficientemente por humanos e por isso existem ferramentas automatizadas que nos ajudam a alcançar esse objectivo.

Este estudo propõe uma ferramenta automatizada de modo a resolver o problema com a dispersão de informação. Desenvolvendo assim uma aplicação baseada nestas ferramentas capazes de aceder à internet, recolher e organizar grandes quantidades de dados sobre as freguesias portuguesas.

1.1 Estado da arte

Procurar actualmente por informação na *internet* é considerado um acto trivial e relativamente fácil com a ajuda de motores de busca, mas para que isto seja possível, existe uma complexa multitude de ferramentas de procura automática.

Para analisar o problema em detalhe, várias ferramentas têm de ser compreendidas e estudadas como o "*web crawler*", uma das primeiras ferramentas a surgir que serve basicamente para realizar uma colecção de endereços e, em ferramentas mais avançadas, uma colecção de páginas de modo a criar um índice de toda a informação, que é a base de como os motores de pesquisa funcionam (Olston & Najork, 2010). Outra importante classe de ferramentas em estudo são os "*internet bots*" que nos ajudam a gerir grandes quantidades de dados. Por exemplo, estes robôs computacionais estão largamente aplicados na Wikipédia desde que esta enciclopédia *online* se tornou um sistema de dados centralizado e por isso com a sua constante edição advém uma constante supervisão (Wikipedia: Bots, 2013). A última ferramenta a estudar serão os "HTML *parsers*", um processo de análise de textos contendo código e informação com o objectivo de extrair apenas os dados desejados. No Capítulo 2 todas estas ferramentas serão revistas em mais pormenor.

1.2 Objectivos do estudo

O objectivo principal deste estudo como o nome do relatório indica é o desenvolvimento de uma ferramenta automatizada para consolidar a informação sobre freguesias portuguesas. Quando se diz consolidar, é no sentido de reforçar a informação existente sobre freguesias portuguesas para um utilizador que necessite deste tipo de dados. Este reforço passa por uma actualização e introdução de novos dados específicos de cada freguesia que não se encontrem directamente disponíveis ou então que a sua pesquisa e recolha seja bastante trabalhosa. Por isso o que se deve realmente realçar neste objectivo principal é a sua automatização, que permite retirar uma grande carga de trabalho tedioso ao utilizador. Por exemplo, sem a automatização para um utilizador pesquisar e recolher dados demográficos, geográficos e contactos devia aceder a pelo menos três bases de dados diferentes para depois procurar em cada uma os dados da freguesia. Mas e se forem todas

as freguesias de um concelho? O trabalho para um utilizador cresce exponencialmente tornando a automatização deste processo numa salvação.

Para criar esta ferramenta automatizada é essencial compreender as ferramentas já existentes dentro deste campo assim como quais bases de dados a utilizar. Deve definir-se as bases de dados *online* e *offline* com informação relevante sobre freguesias e disponibilizar ao utilizador a escolha das mesmas para que o processo de pesquisa e recolha corresponda àquilo que o utilizador pretende. Entretanto, uma análise comparativa da estrutura das páginas sobre freguesias da Wikipédia é realizada com este mesmo intuito, perceber os dados que devem ser consolidados. Como a Wikipédia também ficará com freguesias desactualizadas, existe uma proposta para uma estrutura uniforme das páginas sobre freguesias da Wikipédia, tendo em conta as capacidades desta ferramenta.

Existem três ferramentas que devem ser estudadas meticulosamente para que em sequência se possa desenvolver o programa de recolha de dados: “*Web crawler*s”, “*internet bots”* e “*HTML parsing”*. A primeira questão a resolver é a definição de uma solução de *software* que permita desenvolver um *internet bot* capaz de recolher os dados de um *website* específico e associar esses mesmos dados a dados de bases de dados *offline*. Em segundo lugar é necessário desenvolver uma ferramenta de análise das páginas web com base em *HTML Parsing* para concretizar a recolha de informação das bases de dados *online*, permitindo a separação dos dados importantes para o utilizador do código irrelevante na pesquisa. A terceira questão a resolver é a acessibilidades às várias plataformas e diversos *websites* para que o programa possua uma ampla base de dados, desenvolvendo um método com base em *Web Crawling* para aceder a várias bases de dados *online*.

Depois de estudadas e antes de serem implementadas estas ferramentas, é estudado o processo de engenharia de *software* porque como se trata de um programa relativamente avançado, todas as etapas devem estar bem planeadas. O desenvolvimento da ferramenta, passará pelas etapas da engenharia de *software*: especificação de requisitos, desenho, implementação e testes, assegurando assim que se cumprem todos os objectivos.

1.3 Plano de trabalho

Este trabalho foi realizado por etapas, mais propriamente por métodos que conduzem a realização de uma pesquisa até a um desenvolvimento. A primeira etapa consistiu numa revisão bibliográfica constituída pelo que existe até à data sobre *internet bots*, *web crawlers* e *html parsing*. O objectivo desta primeira etapa foi aprofundar os conhecimentos teóricos sobre estes temas de forma a poder aplica-los posteriormente no trabalho.

Na segunda etapa, procedeu-se o desenvolvimento formal da ferramenta automática. Nesse contexto, foi estudado primeiramente o processo que uma aplicação sofre desde a sua criação até esta estar completa, nomeadamente a engenharia de *software* e todos os passos associados. Estes processos contam numa primeira fase com a especificação dos requisitos da aplicação, de seguida o desenho do programa, a implementação e por fim os testes.

Por fim é realizado este relatório de forma a verificar a viabilidade do modelo proposto.

1.4 Estrutura do relatório

Este trabalho está organizado em seis capítulos. No Capítulo 1, "Introdução", é desenvolvido e apresentado o problema em questão e possíveis formas de solucionar este problema apresentando o estado da arte. De seguida, o Capítulo 2, "Estudo e Análise das Ferramentas", aprofunda o estado da arte inicial, providenciando uma maior organização do tema que se subdivide em três subcapítulos, *web crawlers, internet bots* e *html parsing*. No Capítulo 3, "Metodologia e Desenvolvimento da Aplicação", é apresentada a metodologia usada no desenvolvimento da ferramenta, onde estão definidos os caminhos que se podiam seguir neste desenvolvimento e o que foi de facto seguido. Pode-se ainda encontrar a análise dos requisitos, o desenho, implementação, testes e um tutorial da aplicação. No Capítulo 4, "Conclusões", são apresentadas todas as considerações finais. No Capítulo 5, "Referências", como o próprio nome indica é onde se encontram todas as referências bibliográficas e por fim o Capítulo 6 encontra-se um anexo onde se pode encontrar mais material relevante sobre o trabalho.

2. Estudo e Análise das Ferramentas

Na internet, a quantidade de informação existente é praticamente inimaginável. Essa informação é vulgarmente acedida por utilizadores humanos, mas não somos só nós que acedemos a esses dados. Cada vez mais visitas de programas automáticos de recolha de dados acontecem. Estas aplicações actuam quando existem tarefas simples e estruturalmente repetitivas na internet onde são criadas ferramentas específicas para resolver essas tarefas eficientemente e a um ritmo tão alto que se torna humanamente impossível acompanhar. Muitas destas ferramentas apresentam tarefas especializadas e um nome associado como *Spiders*, *Bots*, *Aggregators* e *Agents*. Como se pode observar na

Figura 1, existe uma hierarquia referente a estas aplicações. Qualquer programa que aceda à internet e que recolha dados de localizações específicas pode-se dizer que é um *Internet Bot*. Quando o programa recebe algum tipo de treino do utilizador para procurar uma informação que pode ter algum tipo de interesse para o utilizador já se denomina de *Agent*. Um A*ggregator* consiste num *bot* capaz de combinar dados de várias páginas da *web* cuja informação é relacionada entre si. As *Spiders* ou *web crawlers* já são especializados em percorrer grandes quantidades de páginas na internet (Heaton, 2002).
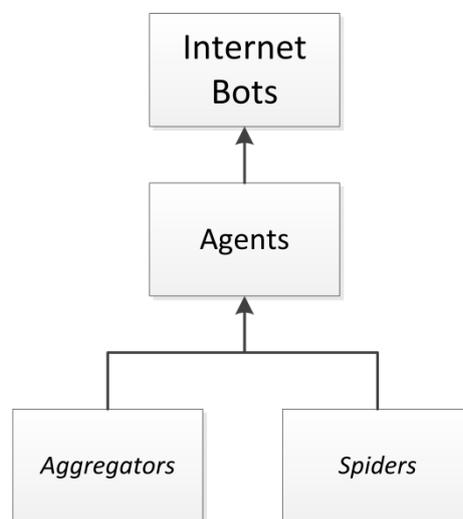
Figura 1- Hierarquia dos Internet Bots

Provavelmente o maior uso destas ferramentas é no chamado *web crawling* que, como se vai ver de seguida, consiste num conjunto de aplicações que procuram, analisam e classificam dados para depois disponibilizar essa informação rapidamente a um utilizador.

2.1 Web Crawlers

A Internet como sistema descentralizado contém um imenso número de conteúdos independentes espalhados por vários servidores. Assim, mesmo que aconteça alguma catástrofe o sistema pode continuar a funcionar sem se perder toda a informação armazenada. Embora esta descentralização seja favorável no aspecto da segurança e do poder que a informação oferece, esta provoca problemas quando se pretende pesquisar qualquer tipo de dados que não se encontrem localmente. A solução encontrada passa por varrer toda a rede e dar a conhecer ao programa onde se encontram os dados. Este sistema que trabalha por trás da internet denomina-se de *web crawler*.

Um *Web Crawler*, também conhecido por *web robot* ou *spider*, é um sistema automático de recolha de páginas de *websites* em massa. Este sistema começa com uma lista de endereços designados de sementes. À medida que o sistema examina as páginas também recolhe e classifica todos os endereços encontrados ao longo da análise de acordo com a relação entre o novo endereço e o endereço semente. Esta relação distingue um endereço interno de um endereço externo ao *website* para que seja possível estabelecer prioridades. Por fim, como se pode ver na Figura 2 o sistema guarda os dados de modo a que um utilizador os possa aceder futuramente (Castillo, 2004).
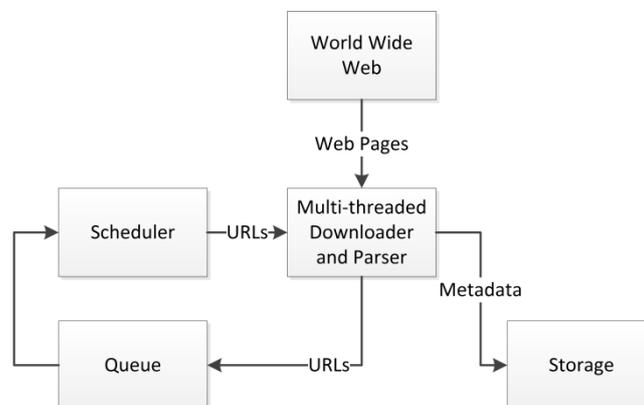
Figura 2 - Arquitectura de alto nível de um Web Crawler

Provavelmente o primeiro *web robot* foi o World Wide Web Wanderer que tinha como função medir o tamanho da *World Wide Web*, tendo também a capacidade de se tornar num motor de busca (Gray, 1996). O *web crawling* é a base de qualquer motor de busca e por causa dos *bots* serem importantes nessa pesquisa foram desenvolvidos para serem mais eficientes com a informação (Brin & Page, 1997). Assim, recolhiam a

informação e indexavam-na para posteriormente poder fornecer a um utilizador uma pesquisa por comparação desses índices (Olston & Najork, 2010, p. 176).

Este tipo de programas resolvem um dos objectivos no trabalho a desenvolver, mais propriamente o acesso a diversos *websites*, para assim conseguir-se alcançar um programa capaz de recolher informações de várias plataformas online. Estas plataformas *online* com a ajuda dos *web crawlers* passam a ser consideradas pelo programa como bases de dados *online*, permitindo assim um acesso a todos os dados públicos desses *sites*.

2.2 Internet *bots*

Um *internet bot* é uma ferramenta automática ou semiautomática que realiza tarefas simples e repetitivas na internet de uma forma muito mais rápida e eficaz do que seria humanamente impossível. Esta ferramenta é normalmente usada de uma forma muito restrita no conceito de número de *sites* a que pode aceder e recolher dados, isto é, são programados para um *site* em específico e só funcionam nesse site em particular (Heaton, 2002, p. 317). Além dos *bots* serem implementados neste tipo de aplicações de uso geral, também podem ser aplicados para situações semelhantes em que é necessário um tempo de resposta extremamente rápido, um tempo de trabalho longo ou uma disponibilidade tal que não seria possível a um humano como por exemplo *bots* de videojogos *online*, em *sites* de leilões ou em salas de *chat* (Arian, pp. 171-173).

No caso de alguns videojogos existem personagens que não são jogáveis, vulgarmente conhecidas por *NPC's* (*Non-player Character*), que se encontra no mundo virtual e actuam como se fossem controlados por um jogador. Este controlo é realizado por um *bot* que por vezes tem algum modelo de inteligência artificial, reagindo a acções e diálogos proporcionando assim ao jogador uma atmosfera mais envolvente.

Os *sites* de leilões assim como as salas de *chat* precisam de moderação da linguagem colocada pelos utilizadores. Esta moderação é realizada através do uso de *bots* capazes de tomar decisões sobre daquilo que está escrito, filtrando assim conteúdo inapropriado e banindo os causadores se continuarem com o mesmo comportamento.

Como os robôs em fábricas aumentam a produtividade de uma empresa, os *bots* foram vistos como uma ajuda relativamente à parte comercial de empresas ligadas à informática. Por exemplo, o Google Inc. está na vanguarda da publicidade na internet com o seu *bot* Google AdSense que tem por base o histórico de navegação do utilizador para

que em páginas onde exista publicidade seja possível apresentar a mais relevante para cada indivíduo, sendo o lucro deste sistema sustentado pela quantidade de cliques ou visualizações de cada anúncio (Ledford, 2008, p. 8).

Os *bots* podem também ser utilizados para fins de caridade, como no projecto FreeRice, que com a ajuda de publicidade na página e respostas correctas por parte dos utilizadores, é capaz de doar uma certa quantidade de arroz para o World Food Programme ajudando a alimentar os mais necessitados. Com isto foram criados *bots* capazes de responder às perguntas de resposta múltipla e assim fazer com que o número de doações fosse largamente maior do que seria de esperar (Arian, pp. 171-173) .

Como o grande objectivo é a criação de um programa automatizado que acede à internet, não podiam deixar de ser estudadas as aplicações dos *internet bots*. Para o trabalho em estudo, isto acrescenta a capacidade de recolher dados de um *website* através da selecção de onde os vais pesquisar e descarregar. Logo à partida é então praticável a pesquisa especifica de dados de uma única freguesia na imensidão de um *website*.

2.2.1 Fins maliciosos

Como existem bons usos para os *bots* também existem maus usos, ou seja, estes são usados com fins destrutivos ou maliciosos. Um dos casos que tem vindo a tornar-se comum é o ataque DOS (*denial of service*), que tem como base uma primeira fase em que o objectivo é infectar um grande número de computadores com o *bot*. Numa segunda fase os computadores infectados enviam um grande número de pacotes ao mesmo tempo a um servidor alvo. O resultado é a falta de resposta do servidor para tantos pedidos, afectando assim a sua disponibilidade (Puri, 2003).

Um outro caso que não foi pelos melhores caminhos foi a fraude em massa que existiu no *eBay* devido a *bots* que faziam licitações de produtos automaticamente ao preço mínimo, não dando assim nenhuma hipótese às pessoas de comprar o produto. Quando isto veio a público, o site começou a ter uma maior actividade de *bots* o que levou a empresa a criar o seu próprio *bot* para combater estes esquemas (Arian, pp. 171-173).

Outro problema surgiu assim que apareceram as primeiras publicidades na internet que geravam lucro a partir do número de cliques. Um *bot* pode fazer uma fraude por cliques o que criava grande transtorno entre as empresas de publicidade e os donos das páginas web. Por vezes era necessário retirar essas publicidades porque estes ataques

tinham grande impacto monetário (Knight, 2005). Existem também alguns usos menos polémicos mas que também afectam o sistema atacado como os *bots* para videojogos *online* do tipo *Massive Multiplayer Online Role Playing Game* (MMORPG) que retiram o esforço do utilizador de procurar por recursos, estragando completamente uma economia virtual.

Os *Spambots*, que como o próprio nome indica, procuram endereços electrónicos em várias páginas da Internet e de seguida enviam grandes quantidades de informação ou esquemas maliciosos como o *phishing.*

Podem também ser encontrados *bots* que fazem *downloads* de *web sites* completos, o que afecta negativamente a largura de banda do servidor, ou então com esses dados criar um *web site* semelhante sem a autorização do autor (Arian, pp. 171-173).

2.2.2 Técnicas "*anti-bot*"

Por vezes estes recursos computacionais são explorados e usados de forma imprópria como por exemplo na criação de *spam* ou ataques à disponibilidade a sites. Para combater estes casos, novas ferramentas automatizadas tiveram de ser criadas para controlar este uso indevido. Um exemplo de um caso de uso incorrecto de internet *bots* era a criação de contas de correio electrónico em massa, contas Youtube para efectuar *spam* de esquemas fraudulentos para assim enganar os utilizadores e ganharem dinheiro ilegalmente ou ataques à disponibilidade através do uso dos campos de pesquisa de um *website* (Ahn, Blum, Hopper, & Langford, 2003).

No combate a estes esquemas a ferramenta mais utilizada é o CAPTCHA ("*Completely Automated Public Turing test to tell Computers and Humans Apart*"). Esta ferramenta baseia-se numa imagem que só pode ser compreendida por humanos, isto é, aparece uma imagem de tal maneira alterada que nenhum programa consegue ler o que lá está escrito e assim o utilizador tem de escrever a palavra que lá estiver confirmando assim que não é um computador.

Mais recentemente, este *software* em alguns casos passou a integrar duas imagens em que uma é igual à que estava em vigor e a outra nova pertence a um livro ou imagens/fotográfias de forma a identificar o que lá está escrito. Denomina-se reCAPTCHA e tem como objectivo ajudar a digitalizar livros, jornais e programas de rádio mais antigos. Como cerca de 200 milhões de CAPTCHAs são resolvidas todos os dias por humanos,

decidiram canalizar esse esforço para algo extremamente positivo, tornar estes puzzles em livros que realmente se possam ler (Google Inc, 2013).

2.2.3 Análise de *bots* da Wikipédia

Para manipular uma grande quantidade de dados de forma autónoma e eficaz, no ano de 2002, foi criado um dos primeiros *bots* da Wikipédia chamando-se Rambot (Ramsey, 2010). Este *bot* tinha uma simples mas muito útil tarefa que se baseava numa base de dados dos Censos dos Estados Unidos da América e em curtas frases que seriam preenchidas com essa informação sobre as localidades. Com este desenvolvimento, ao serem geradas páginas de virtualmente todos os locais possibilitou uma expansão da informação por parte dos utilizadores.

Os *bots* são usados para simplificar e ajudar os humanos a controlar e organizar grandes quantidades de dados, mas com algumas alterações estes podem danificar irremediavelmente uma quantidade enorme de páginas antes que fosse detectado por um humano. Com este problema, criaram-se categorias e regras de utilizadores independentes para este caso particular.

Para controlar eficazmente o trabalho dos *bots* na Wikipédia foi necessário criar políticas específicas para verificar o seu funcionamento, prevenindo assim possíveis erros e problemas. Os novos *bots* passam por uma fase de aprovação onde o operador deste tem de demonstrar que ele é inofensivo e útil para a comunidade. Além disso o operador tem de garantir que o *bot* não consome recursos desnecessariamente, por isso este tem de ser eficiente realizando apenas tarefas que são previamente aceites pela comunidade. Para complementar este sistema, o utilizador deve explicar na página inicial do *bot* todas as suas especificações, isto é, todos os detalhes acerca das tarefas que executa, se este processo é assistido ou se é completamente automático e qual o ritmo a que opera. Depois da aprovação o *bot* vai ser testado na realidade sendo constantemente monitorizado para assegurar que trabalha conforme o estipulado. Passando este último teste é deixado a cargo do operador que criou o *bot* (Wikipedia: Bot policy, 2012).

Actualmente na Wikipédia existem quase 1500 *bots* (Wikipedia bots, 2012), mas apenas cerca de 240 estão sempre activos (Wikipedia bots by status, 2012). Os *bots* mais importantes do ponto de vista da organização são os *adminbots* que, como o próprio nome indica, têm privilégios de administração e por isso podem editar e apagar páginas

livremente fazendo também com que a Wikipédia em geral seja menos espaçosa e mais organizada. Um desses *bots* chama-se CydeBot e faz o processo de mover e apagar categorias da Wikipédia, actualizando depois as páginas de listas dessas categorias. Presentemente este *bot* já realizou perto de três milhões de edições na Wikipédia (Weys, 2011). Outro *bot* extremamente importante é o ClueBot-NG, que tem a tarefa de detectar e reverter vandalismo automaticamente. Este *bot* tem a capacidade de aprender e assim descobrir vandalismo mesmo que as frases em questão sejam verdadeiras (Carter, 2010). Provavelmente o feito mais conhecido do ClueBot-NG foi verificar que foi realizado vandalismo na página de supremos tribunais, com uma frase que por si só é verdadeira mas se encontra fora do contexto. "O pénis é o órgão sexual masculino", foi a frase descoberta e apagada numa questão de segundos (Nasaw, 2012).

2.3 HTML Parsing

Todo o tipo de programas automáticos que navegam na internet deparam-se com a necessidade de manipular diversos tipos de dados. Para além de ter de assegurar a recolha destes dados, os programas têm de os saber interpretar. Os programas ou métodos que realizam este objectivo denominam-se por "*parsers*", isto é, *softwares* capazes de analisar e distinguir dados importantes de dados irrelevantes. Neste caso, estes métodos distinguem os dados importantes do código HTML como, por exemplo, na necessidade de construir um programa capaz de recolher todas as imagens de um website é essencial possuir um analisador de HTML apto a examinar todos os endereços do *site* contidos no código (Heaton, 2002).

No caso deste estudo, o HTML *parsing* adiciona uma outra componente vital no processo de selecção de dados. Como é um processo automático, só necessita de saber o que é que o utilizador considera importante, isto é, todas os dados relacionados com freguesias. Assim, é possível pesquisar e recolher apenas os dados que são importantes ao utilizador.

3. Metodologia e Desenvolvimento da Aplicação

Uma lista de tarefas necessárias para recolher e aplicar informação sobre os processos constitui uma metodologia. Uma metodologia consiste num conjunto de passos que são seguidos e completos. A metodologia associada à engenharia de *software* demonstra vários processos de *software;* modelos abstractos que por vezes podem ser modificados de forma a adaptarem-se a um desenvolvimento específico (Smith & Sarfaty, 1993). Os três modelos genéricos apresentados são: o modelo incremental, o modelo orientado ao reuso de *software* e o modelo em cascata.

No modelo incremental de desenvolvimento de software são intercalados três processos: especificação de requisitos, desenvolvimento e validação. Dito por outras palavras, o sistema é desenvolvido como uma série sucessiva de novas versões. Esta metodologia possui diversas vantagens para a maioria das empresas, comércio *online* e sistemas pessoais visto reflectir de forma mais natural a resolução de problemas. Cada novo incremento ou nova versão do sistema integra novas funcionalidades, proporcionando assim uma visão rápida e precoce daquilo que o sistema pode fazer de acordo com o estipulado. Por outro lado, o progresso do desenvolvimento nesta metodologia não é facilmente mensurável visto que, como são criadas novas versões do sistema rapidamente, também têm de ser concebidos documentos para cada versão, o que frequentemente não acontece pois contribui, de forma significativa, para tornar o procedimento dispendioso. Para além desta desvantagem, o sistema tende a degradar-se à medida que novos incrementos são efectuados, tornando-se num grave problema quando os sistemas são maiores e mais complexos (Pressman, 2010).

Um outro modelo genérico existente para o desenvolvimento de *software* é o modelo orientado ao reuso de *software.* Basicamente, o que acontece neste modelo é uma abordagem ao desenvolvimento do sistema com base em componentes já criados ou existentes. Este processo de desenvolvimento foca a incorporação destes componentes no novo sistema sem ter de começar esse novo sistema de raiz. Este modelo acontece frequentemente mas de maneira informal, pois na maior parte dos projectos existe sempre algum tipo de reuso de componentes, nem que por vezes tenha que se fazer pequenas modificações ao mesmo. Quando um sistema se baseia inteiramente neste modelo de desenvolvimento existem quatro processos diferentes dos outros modelos: a análise dos componentes, a modificação de requisitos, desenho do sistema com o reutilizado e o desenvolvimento e integração. Uma grande vantagem deste modelo é a quantidade

reduzida de desenvolvimento de novo *software*, embora frequentemente alguns requisitos tenham de ser inevitavelmente comprometidos (Basili, 1990).

Por fim o modelo em cascata que, como o próprio nome indica, segue de uma fase para a outra servindo de exemplo abstracto como um processo orientado a um plano, isto é, deve ser previamente planeado todo o tipo de processos e actividades antes de se dar início a qualquer tarefa. Os principais passos deste modelo são: a análise e definição de requisitos, o desenho do sistema, a implementação e testes de unidade, integração e testes de sistema e por fim operação e manutenção. Como a documentação é produzida no fim de cada fase, é possível monitorizar o progresso do programa sobre o plano desenvolvido, mas como cada fase é realizada independentemente das outras, leva a uma inflexibilidade e é consequentemente difícil mudar por exemplo algum requisito depois de essa fase já ter sido concluída.

Dado tratar-se de um projecto de reduzida dimensão, o desenvolvimento da ferramenta de *software* associada a este trabalho segue este modelo em cascata. Como referido, este modelo dita que todas as etapas e actividades do projecto são planeadas antecipadamente e por isso a progressão do trabalho é medida através deste planeamento. O modelo em cascata é um processo de desenvolvimento de *software* que em teoria considera que uma actividade tem de estar terminada para prosseguir para a próxima, mas na prática este modelo raramente é linear o que implica um determinado *feedback* entre actividades. Este modelo foi alterado de modo a integrar-se no desenvolvimento deste *software*, como é possível visualizar na Figura 3, contando assim com quatro actividades fundamentais (Sommerville, 2010):

1. Análise de requisitos, onde todas as funcionalidades e restrições da operação do *software* são definidas;
2. Desenho, que envolve identificar os componentes fundamentais do sistema e seus relacionamentos;
3. Implementação, onde se converte o desenho para um programa de computador;
4. Testes, para assegurar que o programa faz aquilo que é suposto fazer de acordo com os requisitos de *software*.
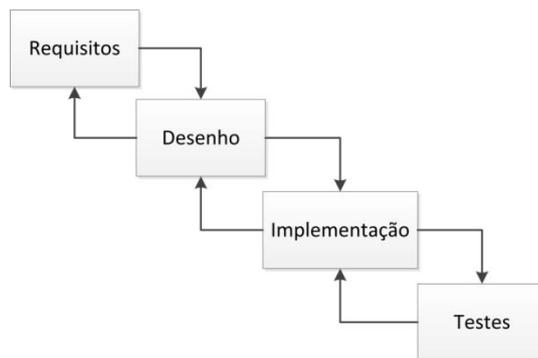
Figura 3- Diagrama do modelo em cascata usado

As seguintes quatro secções descrevem com detalhe estas quatro actividades. Posteriormente é apresentado um manual de utilização da aplicação desenvolvida.

3.1 Análise de requisitos

A análise de requisitos do sistema é a base de todo o projecto e por isso todas as escolhas feitas são críticas para o seu sucesso. Os requisitos estão divididos em requisitos funcionais e requisitos não funcionais. Enquanto os requisitos funcionais definem o que é que o sistema pode ou deve fazer, os requisitos não funcionais apresentam restrições ao funcionamento do sistema, sendo frequentemente aplicados a todo o *software* (Pressman, 2010). Tendo em conta esta informação, o requisito não funcional mais importante desta ferramenta é que este sistema deve ser fácil e prático de usar, tanto para utilizadores básicos como para utilizadores avançados. Por outro lado, os requisitos funcionais deste programa incluem:

- O sistema deve compilar dados tanto de plataformas *online* como de plataformas *offline* de acordo com cada freguesia;
- Os dados de uso *offline* devem ser aceites como folha de cálculo (mais especificamente ficheiros ".xls");
- O programa deve conter algum tipo de personalização da entrada dos ficheiros para que mesmo quando a folha de cálculo possua uma tabela com uma diferente ordem de colunas seja possível funcionar;
- Deve existir uma secção de gestão destas folhas de cálculo, para adiciona-las ou elimina-las e para definir quais as colunas de saída;
- A selecção dos distritos, concelhos e freguesias portuguesas deve existir numa forma que permita a um utilizador selecionar várias freguesias do

mesmo concelho ou selecionar um concelho e automaticamente todas as freguesias desse concelho são seleccionadas;

- Para aquisição de dados *online* o sistema deve aceder a sites oficiais e com conteúdos fidedignos;
- Os dados adquiridos *online* podem possuir contactos das freguesias, tais como números de telefone, morada e e-mails quando existirem;
- Pode ser incluído um modo de escolher as bases de dados (folhas de cálculo) de saída de entre as diversas bases de dados de entrada;
- Deve existir um simples editor de texto onde seja possível incluir variáveis relacionadas com as bases de dados;
- A maioria das personalizações podem ser automaticamente gravadas para aumentar a eficiência e manter a integridade do sistema, permitindo ao utilizador voltar a usar a aplicação no estado onde a deixou;
- Os dados de saída devem aparecer para o utilizador numa tabela;
- Os dados de saída devem ser exportados para ficheiros de texto ou ficheiros de folhas de cálculo;
- O texto de saída deve incluir variáveis que são posteriormente substituídas pelos respectivos dados de cada freguesia;
- Os dados de saída devem ser gravados de várias maneiras, como por exemplo toda a informação num único ficheiro ou um por cada freguesia, um por cada concelho ou um por cada distrito;
- Quando o sistema está a recolher e a organizar os dados, um indicador visual do progresso pode ser usado para representar quanto progresso já foi feito e quanto falta fazer.

Mesmo possuindo uma boa lista de requisitos não significa que estes não sejam afectados até ao final do projecto, pois é normal existir mudança conforme o sistema evolui. A actividade seguinte é o desenho do *software*, onde se começa a interligar as diferentes partes do sistema e eventualmente problemas relacionados com os requisitos podem surgir. Como na prática o modelo em cascata não é linear, existe um forte feedback entre estas duas actividades devido às possíveis mudanças para consertar os problemas de interligações.

3.1.1 Dados e bases de dados

A selecção das bases de dados que se podem utilizar está separada em dois grupos: as que são acedidas através da internet e as que estão em ficheiros guardados no computador (*online* e *offline*, portanto). As bases de dados que estão na internet são as que englobam dados referentes a moradas, contactos, monumentos e até alguma possível descrição histórica das freguesias. Estes endereços são:

- Para recolha de moradas, *e-mails*, página web e contactos, tais como telefone ou fax, existe o site da Associação Nacional de Freguesias (ANAFRE) — www.anafre.pt
- Para recolha de dados estatísticos que abrangem toda a situação social em Portugal pode aceder-se à página da PORDATA (www.pordata.pt) onde se podem encontrar dados sobre a população, saúde, educação, emprego e condições de trabalho, produto, rendimentos e níveis de vida, habitação, conforto e bem-estar, segurança social, cultura, justiça, contas nacionais e função social do estado, empresas e trabalhadores, entre muito outros. Embora esta organização possua uma grande quantidade de dados, estes só são específicos até ao nível dos concelhos.
- Para a recolha de gentílicos relativos a freguesias pode ser consultada uma página da Wikipédia que contém uma lista com mais de 1200 nomes (Lista de Gentílicos de Portugal, 2013).
- Um *site* semelhante ao da ANAFRE é o Portal das Freguesias (www.freguesias.pt) onde, para além de se encontrar praticamente todos os dados que existem na ANAFRE, também possui a imagem da insígnia da freguesia e os nomes da classe administrativa e executiva da freguesia, tais como presidente, secretário, tesoureiro, etc. Por vezes também se pode encontrar uma breve descrição da freguesia e também um mapa que destaca a freguesia dentro do concelho.
- Na página *web* do Instituto de Gestão do Património Arquitectónico e Arqueológico (www.igespar.pt) pode-se pesquisar todo o tipo de património cultural, embora este seja específico só até aos concelhos, não contendo dados únicos e referentes a freguesias.

Quanto às bases de dados *offline* que se encontram em ficheiros do tipo folhas de cálculo com tabelas de dados foram encontrados três ficheiros determinantes:

- Uma folha de cálculo com todos os distritos, concelhos, freguesias e códigos de freguesia de Portugal, realizada pela Autoridade Tributária e Aduaneira (ATA) que pode ser usada como tabela relacional.
- Outra tabela importante relativa a dados geográficos é a Carta Administrativa Oficial de Portugal (CAOP), produzida pelo Instituto Geográfico Português.
- A maior fonte *offline* vem do Instituto Nacional de Estatística, mais precisamente a versão *offline* dos Censos 2011, que é uma enorme compilação de dados demográficos.

Apesar de existirem bastantes bases de dados, apenas algumas podem ser escolhidas visto que podem não ser relacionadas com freguesias, terem de ser excluídas devido a redundância de dados entre elas ou então não serem oficiais e por isso não fidedignas. Posto isto, as bases de dados escolhidas são: o *site* da ANAFRE, a lista de gentílicos da Wikipédia, a tabela relacional da Autoridade Tributária e Aduaneira, a Carta Administrativa Oficial de Portugal e os Censos 2011. Como foi estipulado nos requisitos, as tabelas *offline* escolhidas podem ser actualizadas e modificadas conforme o uso do sistema.

3.1.2 Análise da estrutura de páginas sobre freguesias na Wikipédia

A Wikipédia também ajuda na decisão de quais dados e bases de dados que se podem utilizar, pois todas as freguesias portuguesas têm página criada, umas com mais informações e outras com menos. Isto verifica-se pela cor dos endereços para essas páginas na lista das freguesias, pois quando está escrito a azul existe página e quando está a vermelho não existe (Lista de Freguesias de Portugal, 2013). As que contêm menos informação, normalmente têm só o essencial como por exemplo o número de habitantes, a área da freguesia e as que têm mais informação incluem património cultural como a sua história e fotografias.

Como é óbvio, quanta mais população e mais área uma determinada freguesia possua, maior a probabilidade de esta ter algum tipo de património cultural ou natural

importante. Também é importante estabelecer a relação dessa freguesia com a sua localização pois existem grandes diferenças entre o interior do país e o litoral.

Um exemplo de uma localização dependente do turismo são as regiões autónomas da Madeira e dos Açores. Em comparação com zonas não dependentes, praticamente em todas as freguesias das regiões autónomas da Madeira e dos Açores, verifica-se um aumento da quantidade da informação, indo desde património a descrições dos mesmos incluindo fotografias. Este aumento observa-se porque um dos meios de obtenção de riqueza das ilhas é baseado no turismo e o próprio governo das ilhas, principalmente no Arquipélago da Madeira, cria projectos para aperfeiçoar estes canais de informação.

Quanto à informação disponibilizada, as páginas de localidades em que existe bastante informação seguem um modelo base que inclui muitas das vezes tópicos sobre a história, geografia, economia, transportes, cultura e toda a parte demográfica. Quanto às localidades de menor dimensão, onde existe substancialmente menos informação, é disponibilizada normalmente apenas informação demográfica.

3.1.3 Proposta de página modelo

Grande parte das páginas sobre freguesias portuguesas na Wikipédia não estão em conformidade com a política de edição que embora diga que versões iniciais e/ou incompletas de artigos são aceites mesmo contendo pouca informação, estes são marcados como esboços devido a esta mesma falta de informação. Por isso, para complementar esta falha, vai ser possível com a ajuda deste programa reforçar o modelo de dados existente.

Nos parágrafos anteriores foram apresentadas quais as bases de dados que se podem usar e quais as que já existem na Wikipédia para que agora seja possível fazer a consolidação entre elas. Com base nisto e nos requisitos do programa, pode-se propor uma estrutura uniforme para as páginas das freguesias com base no programa.

A Figura 4 ilustra como é necessário ter especial atenção à forma como a gramática é escrita para que as preposições e outros ligadores se adeqúem ao género da freguesia, concelho e distrito. Uma dica será fazer todas as freguesias concelho a concelho, para que a gramática entre o concelho e o distrito não mude. Com isto, o utilizador pode ainda acrescentar mais informações de outras bases de dados de forma a completar esta proposta pela substituição de variáveis no texto assim como a adição de código específico da Wikipédia, como hiperligações, tabelas ou capítulos.

```
==[var:Freguesia]==
[var:Freguesia], freguesia do concelho da [var:Concelho] e do distrito do
[var:Distrito], tem [var:População Residente-HM] habitantes para [var:Edifícios]
edificios, espalhados por uma área de [var:AREA_20121(Ha)] hectares.

==Contactos==
Para mais informações pode dirigir-se à morada da junta de freguesia: [var:Morada].

Ou então entrar em contacto pelo telefone: [var:Telefones], Fax: [var:Fax] ou E-
mail: [var:E-Mail].

Pode ainda visitar o sítio da internet da freguesia em [var:Página Web].
```

Figura 4 - Proposta de uma estrutura simples e uniforme com base na ferramenta

Para um bom artigo completo sobre uma freguesia, já seria necessário criar grandes textos específicos para cada freguesia, como por exemplo a história, caso que o programa é incapaz de fazer. Posto isto, um editor deve apresentar os seguintes itens:

1. Breve introdução
2. Origem do Nome
3. História
4. Geografia
5. Lista dos lugares
6. Localização da freguesia no concelho
7. Demografia
8. Actividades Económicas
9. Religião
10. Feiras, Mercados e outras Festividades
11. Comunicações e transportes
12. Personalidades
13. Associações recreativas, culturais, desportivas e assistenciais
14. Educação
15. Equipamentos Públicos
16. Galeria de fotos
17. Referências
18. Bibliografia
19. Ligações externas

Com esta listagem, um utilizador pode expandir a informação de uma freguesia com o objectivo de completar o artigo com tudo o que existe na freguesia passível de ser importante.

3.2 Desenho

No desenho do sistema os requisitos são organizados de forma a assegurar o alcance de uma etapa em que o *software* é propriamente desenvolvido e por isso é crucial saber como resolver estes problemas (Sommerville, 2010).

A interface do programa, como mostra a Figura 5, é composta por seis passos, cada um contribuindo para produzir uma saída final.
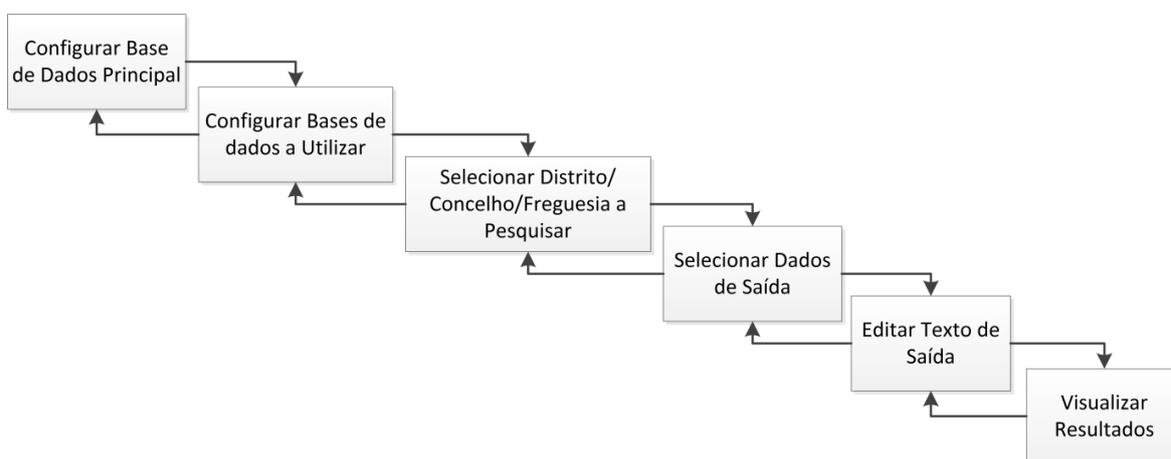
Figura 5- Diagrama de seis passos da interface

A personalização da base de dados principal é realizada usando variáveis da folha de cálculo. Essa folha de cálculo pode ser encontrada na página web da Autoridade Tributária e Aduaneira (ATA, 2010). Como se pode examinar na Figura 6, para o sistema funcionar correctamente uma base de dados principal (com extensão ".xls") é carregada. As colunas definidas para ser possível aceder a esta base de dados são: a coluna dos distritos, a coluna dos concelhos, a coluna das freguesias e por fim a coluna do código das freguesias. Antes de passar ao passo seguinte estas variáveis são gravadas num ficheiro de texto para que seja possível um acesso mais rápido ou continuar onde terminou da última vez num acesso futuro.
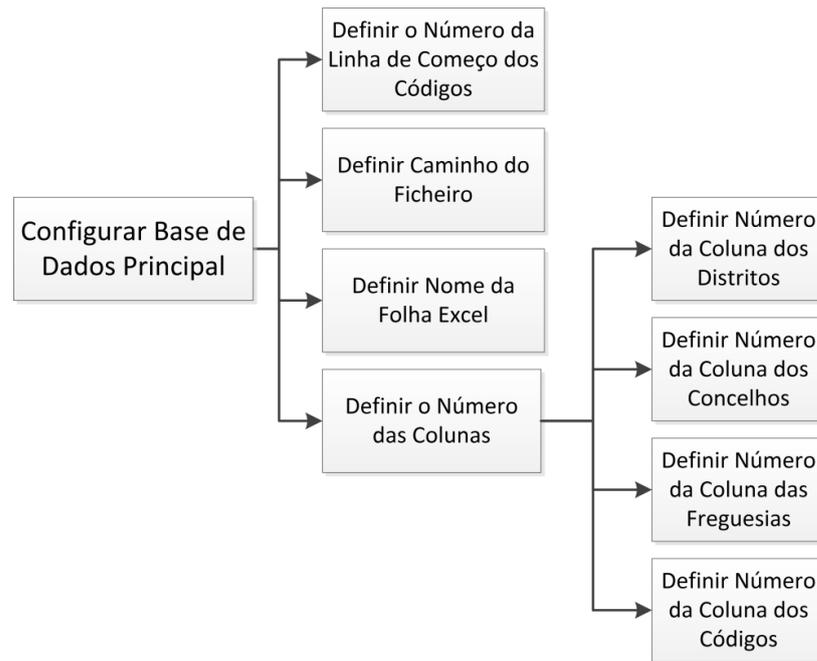
Figura 6- Diagrama de configuração da base de dados principal

Cada freguesia portuguesa possui um código único. Este código é composto por seis digitos e representam a chave primária que este sistema necessita para interligar as bases de dados. O diagrama de interação entre as bases de dados, como mostra a Figura 7, representa um dos requisitos para a interligação de todas as bases de dados *offline*.
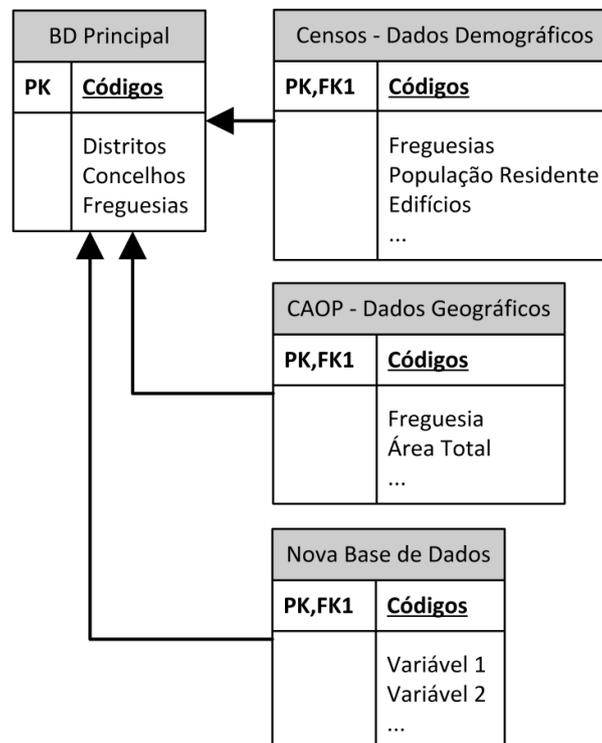
Figura 7- Diagrama de interacções entre bases de dados

Um dos requisitos fundamentais desta aplicação é ser possível actualizar as bases de dados *offline*. Para que isto seja possível, foi introduzido um sistema de gestão para estas bases de dados. Este sistema de gestão foi implementado ao gravar todos os parâmetros necessários para aceder a esses novos ficheiros num ficheiro de texto, onde os dados são separados por ponto e virgula (similar aos ficheiros ".csv"). Estes parâmetros incluem o nome da base de dados, o caminho do ficheiro, o nome da folha de cálculo e o número da coluna do código das freguesias. Para adicionar um novo ficheiro é necessário também introduzir as variáveis presentes nas colunas desse ficheiro. Como o exemplo da Figura 8, isto é realizado a partir da escrita ordenada do nome de cada coluna e de seguida a selecção das variáveis que vão fazer parte da saída do programa. Os ficheiros essenciais para realizar este procedimento podem encontrar-se no Instituto Nacional de Estatística (INE, 2012) ou no Instituto Geográfico Português (IGP, 2012).
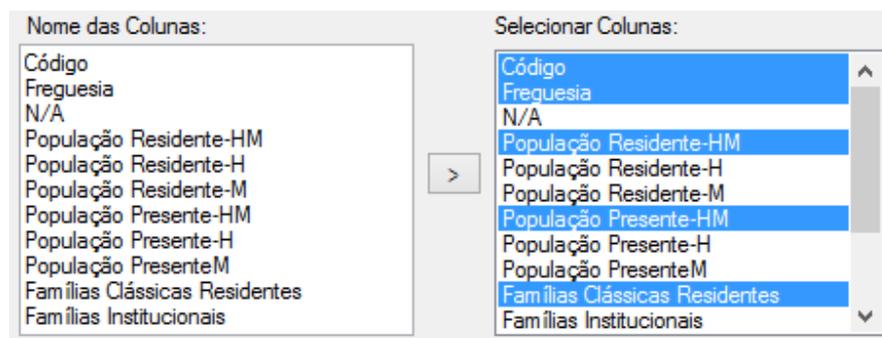
Figura 8- Inserir e selecionar variáveis ao dar nomes às colunas

Para obter algum tipo de resultados é necessário o utilizador escolher antecipadamente a freguesia a pesquisar. Esta escolha é realizada a partir de uma área com três listas provenientes da base de dados principal: uma lista com os distritos, outra com os concelhos e por fim uma lista com as freguesias. A primeira lista de distritos corresponde à maior divisão administrativa. Ao seleccionar um distrito, o utilizador pode escolher de seguida um ou mais concelhos relativos a esse distrito, sendo o mesmo verdade para as freguesias para um determinado concelho. Para efectuar este processo, a base de dados principal é acedida usando uma técnica conhecida por OLEDB: a área geográfica seleccionada pelo utilizador – distrito ou concelho – é pesquisada com o objectivo de filtrar todos os concelhos ou freguesias, respectivamente. O fluxograma apresentado na Figura 9 mostra os passos dados quando o utilizador selecciona um distrito (zona maior) e o sistema procura todos os concelhos (zona menor) pertencentes a esse distrito, sendo o mesmo

conceito aplicado entre concelhos (zona maior) e freguesias (zona menor). Para mais detalhe sobre esta implementação, o código deste método encontra-se no Anexo 1.
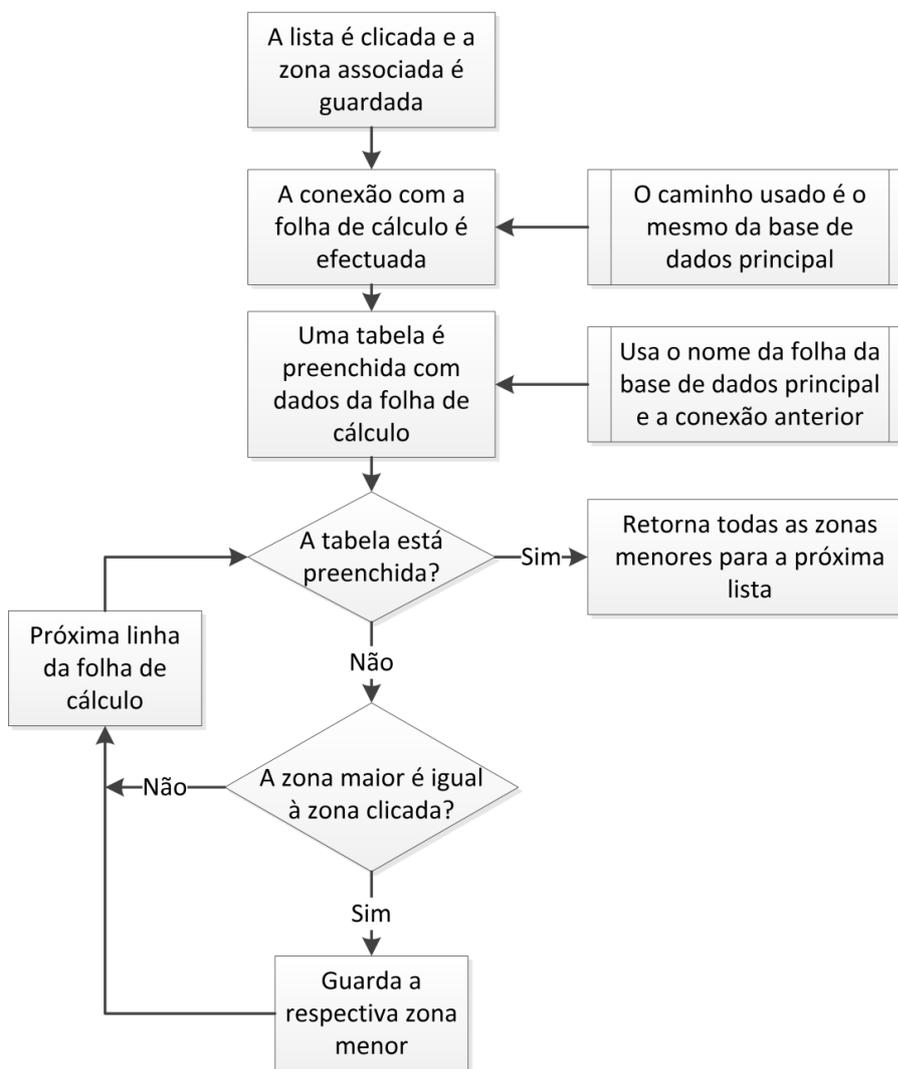
Figura 9- Fluxograma lógico de selecção de zonas

Por razões de eficiência e de simplicidade, a aplicação usa uma técnica que envolve uma edição nos endereços para evitar o uso de um *web crawler*. Como mostra a Figura 10, a página na internet de uma freguesia pode ser acedida ao usar o respectivo distrito, concelho e freguesia, ou então o código de freguesia, simplificando assim a pesquisa por dados. Contudo, devido ao facto da língua Portuguesa usar acentuação, por exemplo para o *website* (www.anafre.pt), o nome de distrito, concelho e freguesia têm de passar por um processo de análise, uma vez que os endereços não contêm acentuação nem espaçamentos. Mais especificamente, os espaços serão convertidos em hífens e é reposta a letra sem acentuação de modo a obter o endereço correcto.

http://www.anafre.pt/distritos/braga/vila-nova-de-famalicao/vila-nova-de-famalicao

Distrito Concelho Freguesia

http://www.freguesias.pt/freguesia.php?cod=031248

Código de Freguesia
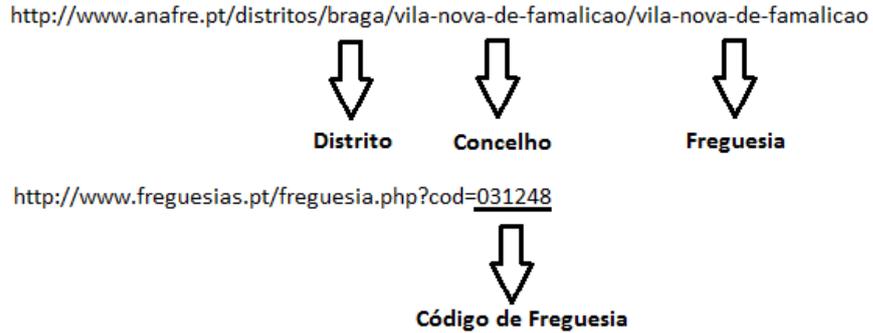
Figura 10- Técnica de pesquisa de dados de freguesia a partir de endereços

Para a recolha de morada, contactos e endereço da página da freguesia são utilizados três métodos para aceder a uma base de dados *online*, mais exactamente no sítio da Associação Nacional de Freguesias (ANAFRE). Na recolha de gentílicos, a base de dados é uma tabela na Wikipédia só de gentílicos de Portugal. Uma outra base de dados a utilizar é um ficheiro com a informação dos Censos 2011. Na recolha da área das freguesias, é utilizada uma base de dados *offline* do instituto geográfico português, mais precisamente, a carta administrativa oficial de Portugal (CAOP). Na Figura 11 podem-se ver as várias interacções entre os métodos e as bases de dados *online* e *offline*.
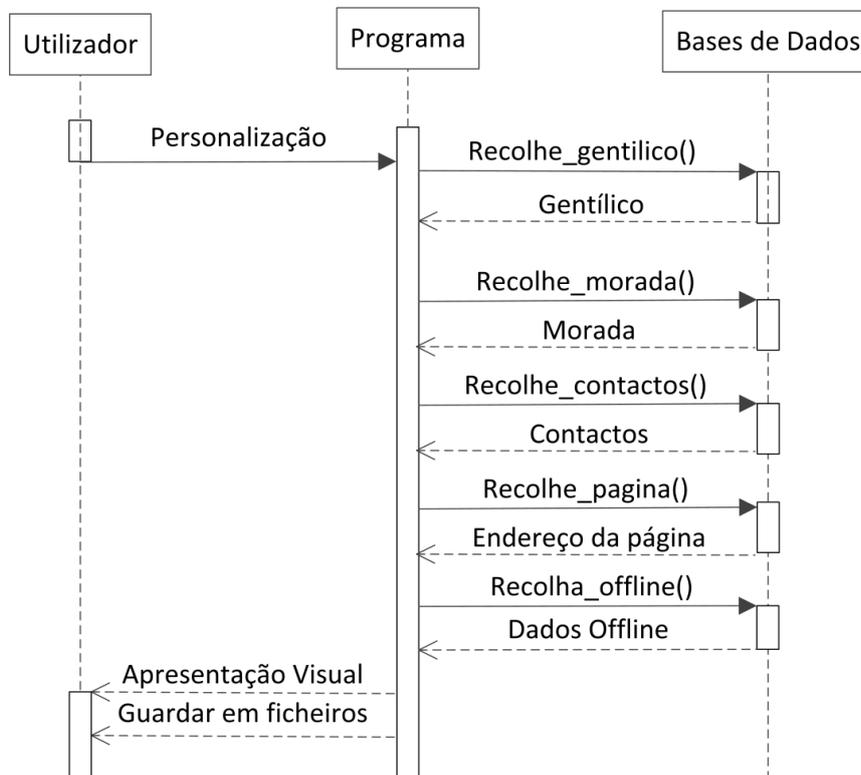
Figura 11 - Diagrama sequencial da aplicação

Para complementar a informação de uma freguesia seleccionaram-se os gentílicos pois estes traduzem-se numa definição dos habitantes da mesma. Gentílicos ou etnónimos são palavras que definem uma pessoa conforme o local onde nasceu ou onde reside. Com o intuito de incluir estas informações, a base de dados usada é da própria Wikipédia e contém cerca de 1200 gentílicos (Lista de Gentílicos de Portugal, 2013). Para isto, como se pode visualizar na Figura 12, foi criado um método que acede à página da internet por meio de uma classe *WebClient*, tendo-se assim acesso ao ficheiro HTML em forma de texto. Assim é possível procurar no texto os dados referentes a cada localidade e retirar o respectivo gentílico. O primeiro passo será a verificação da existência dessa localidade na base de dados, pois nem todas as freguesias possuem gentílico. Isto é executado procurando-se pelo primeiro índice em que esta aparece, para posteriormente poder-se retirar toda a informação da mesma. Esta informação é retirada recorrendo ao próprio código HTML, isto é, ao analisar o texto verifica-se que as Tags em HTML compreendem os dados, sendo assim possível recolhê-los de acordo com a tabela onde se encontram os dados. Estes primeiros elementos recolhidos têm o seu termino no final da linha da tabela, logo as Tags que os circunscrevem são <tr> e </tr> (*table row*). Para limitar ainda mais os dados, separa-se a localidade do gentílico, desta vez pelo meio de outras Tags <td> e </td> (*table data*). De seguida remove-se todas as Tags HTML, código e espaços desnecessários ficando apenas com o que se requeria inicialmente, o gentílico.
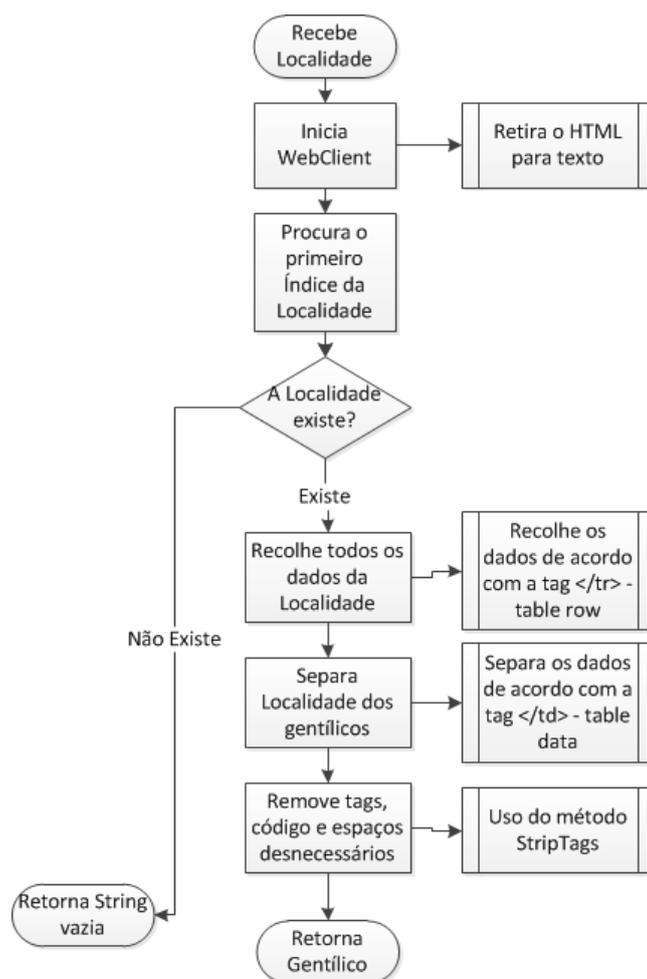
Figura 12- Fluxograma da recolha de gentílicos

Tal como na recolha de gentílicos, a recolha da morada, contactos e página web é realizada através da Internet, encontrando-se na página da associação nacional de freguesias (ANAFRE). Nesta recolha, os dados foram separados em três métodos distintos, pois para além das Tags associadas a estes campos serem diferentes, o tratamento dos dados também o é, mais propriamente a sua existência ou não é mais facilmente tratada desta forma. Neste caso o princípio é o mesmo que nos gentílicos, existindo uma classe *WebClient* que no final retorna o ficheiro de texto do código fonte da página da freguesia. Na recolha da morada, como se observa na Figura 13, procura-se pelo índice da morada para verificar a sua existência. Após esse teste, retira-se toda a informação da morada de acordo com as Tags que definem a lista <dl> e </dl>. Logo de seguida todos os dados são agrupados, pois no código fonte estes estão só separados pelas Tags e , logo como a morada é como um todo, isto simplifica o processo. De seguida são eliminadas as Tags HTML, código e espaços desnecessários.
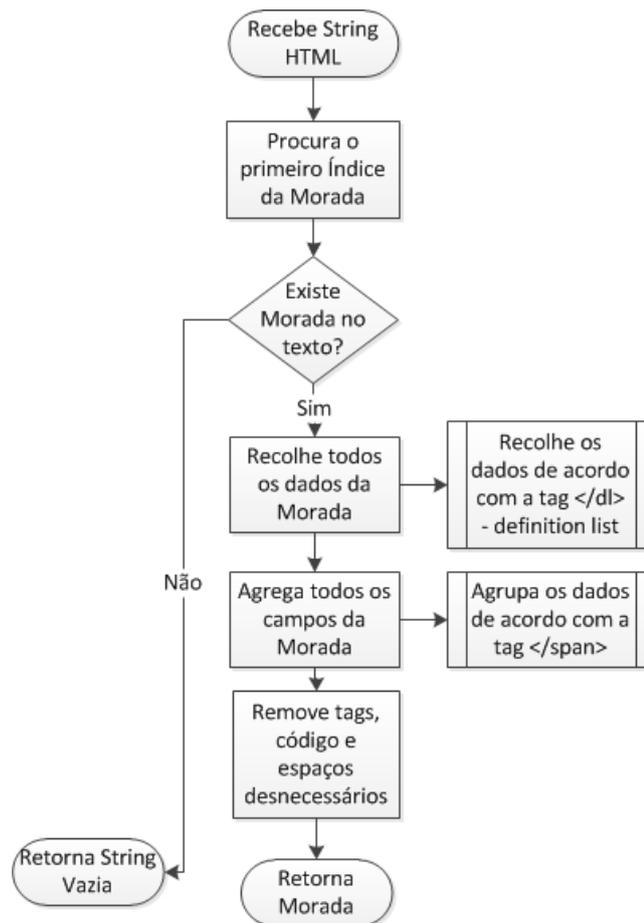
Figura 13 - Fluxograma do método recolhe_morada()

No caso da página web da freguesia, o processo é exactamente o mesmo que na recolha da morada visto que essa recolha só possui uma informação. A única diferença de um método para o outro é a primeira procura que neste método é "Endereço do Sítio".

Quanto à recolha de contactos, como se pode ver na Figura 14, começa-se por avaliar de existem ou não contactos pela procura do índice da ocorrência do mesmo. Se não existirem contactos é retornado um texto vazio, mas se existir continua a pesquisa. Primeiro recolhem-se somente os dados referentes aos contactos com recurso às Tags <dl> e </dl> que se referem a uma listagem de itens, que neste caso são os contactos. Depois para se separarem os dados, verifica-se se existe índice de morada, contactos e Emails individualmente com recurso às tags , e <dd>,</dd>. Ao existirem, os dados já separados são tratados, removendo todas as Tags HTML, código e espaços desnecessários.
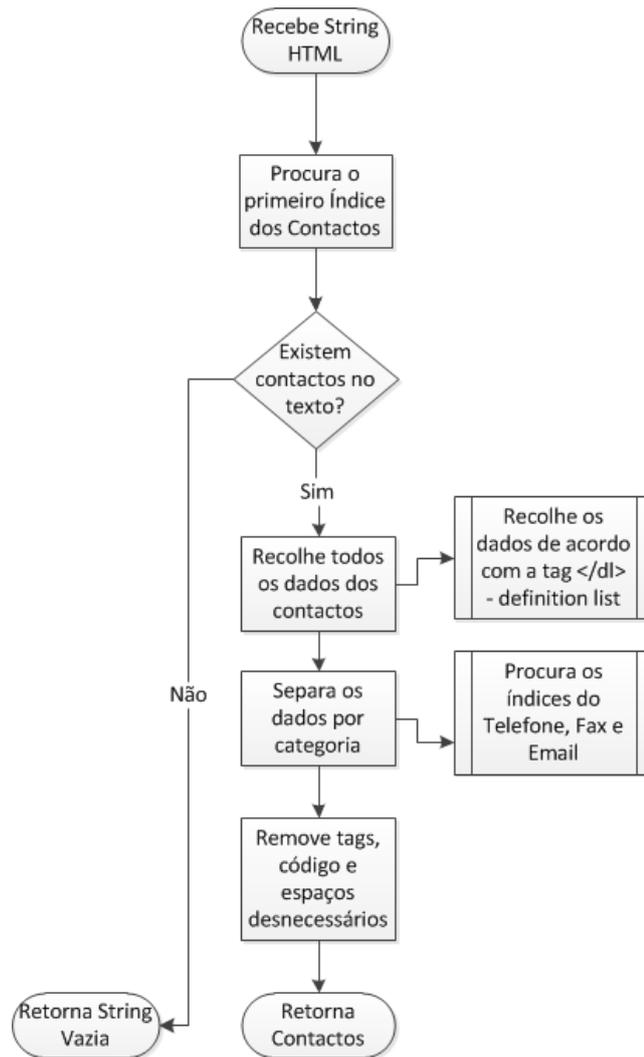
Figura 14- Fluxograma do método recolhe_contactos()

O método que remove o código HTML (*StripTags*), cujo fluxograma lógico é apresentado na Figura 15, é usado por todos os métodos deste sistema quando é necessária uma análise das páginas web. Este método processa o texto tendo em conta os caracteres que delimitam as Tags “<” e “>”. Depois do método receber o texto, este entra num ciclo até que o texto seja todo revisto. Nesta revisão, vai ser verificado se se está dentro de uma Tag e se isto for verdadeiro passa ao próximo caracter senão adiciona ao texto final que vai ser retornado antes de passar ao próximo.
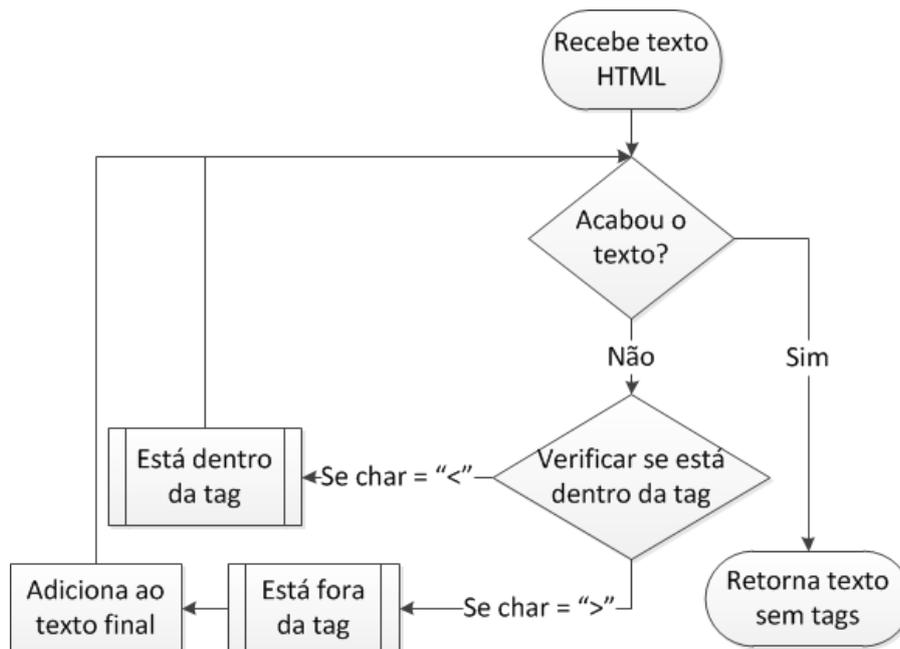
Figura 15- Fluxograma da remoção de código HTML

O passo onde se seleccionam as saídas permite ao utilizador escolher quais os dados que serão processados, mostrados e guardados; por exemplo, quais os dados *online* são recolhidos ou como são gerados os ficheiros de texto finais onde serão guardados os dados. Isto é implementado através da junção de pequenos trechos de texto de cada freguesia de acordo com a introdução feita por parte do utilizador.

O utilizador pode criar um texto modelo ao clicar em variáveis escolhidas previamente no gestor de bases de dados. Ao executar esta acção, um campo ou etiqueta com o nome da variável aparece no texto. A aplicação pode posteriormente alterar estas variáveis para os seus respectivos valores das freguesias e assim um texto baseado num modelo é criado para cada freguesia.

Por fim, uma folha de cálculo é mostrada ao utilizador contendo todos os dados gerados. Esta folha de cálculo é controlada como uma tabela de dados onde as linhas correspondem a freguesias e as colunas são atributos seleccionados pelo utilizado. Como mostra a Figura 16, a construção da folha de cálculo final começa por recolher todos os códigos de freguesia que foram seleccionadas no terceiro passo, ao receber a informação das bases de dados guardadas relativamente às colunas seleccionadas e são adicionados nomes às colunas da folha de cálculo, o que é necessário para a edição de texto. Os nomes de cabeçalho opcionais são os nomes das bases de dados *online* que por serem opcionais sofrem um tratamento especial. Depois disto, as bases de dados *offline* são pesquisadas por dados referentes aos códigos de freguesias previamente recolhidos e consequentemente

adicionados à tabela final. O processo seguinte envolve a pesquisa *online* e consiste nos dados previamente mencionados sobre as freguesias onde são acedidos pela alteração de alguns parâmetros nos endereços. Antes da informação ser apresentada ao utilizador, esta é organizada numa folha de cálculo de acordo com a ordem das bases de dado escolhidas.
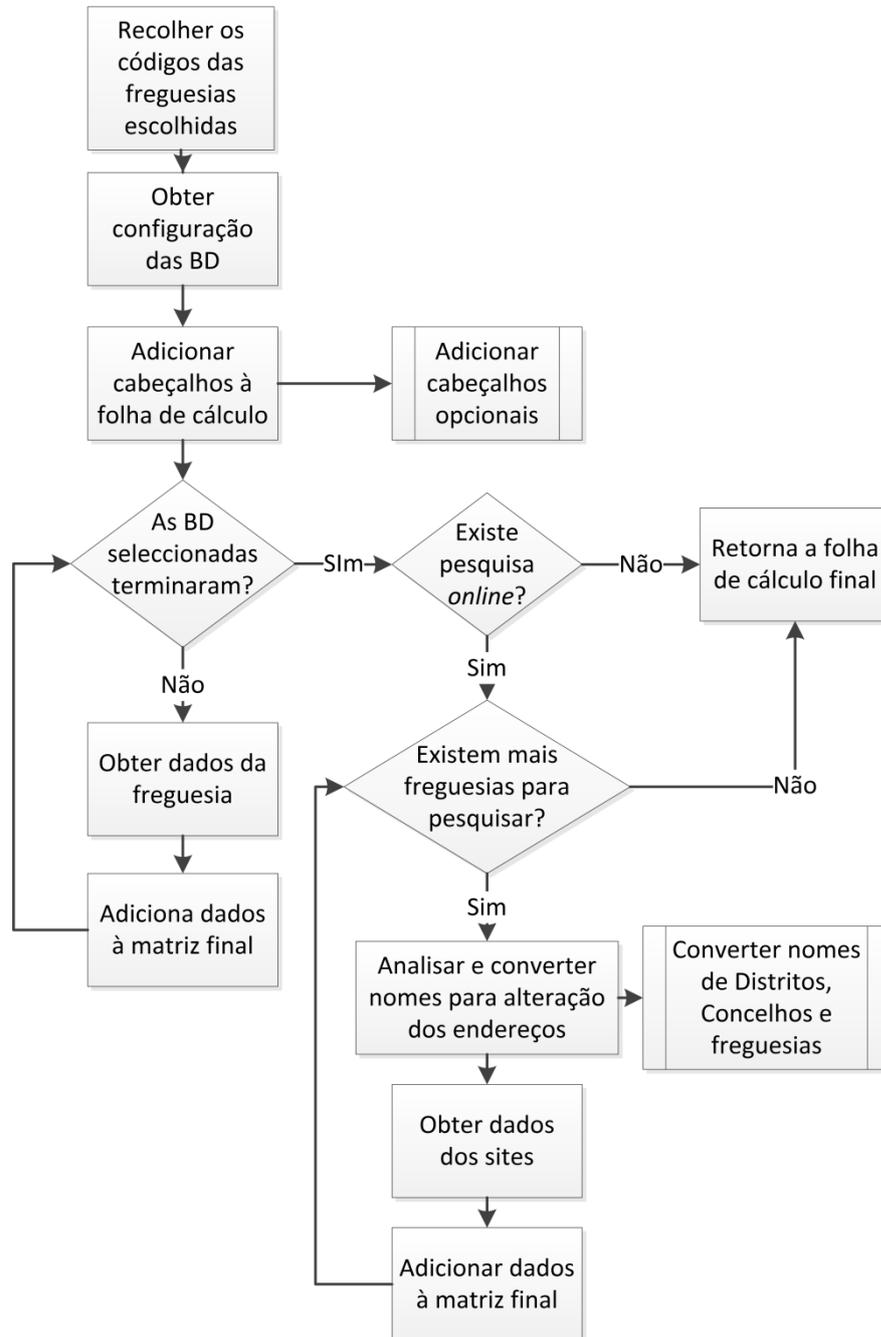
Figura 16- Fluxograma lógico da recolha e organização de dados

3.3 Implementação

Na fase de implementação o desenho da aplicação é transformado em interfaces e código fonte. Visto que o ambiente onde o sistema vai trabalhar é a plataforma Windows, decidiu-se que a melhor linguagem de programação a usar neste caso é a linguagem C#, pois em conjunto com o ambiente de desenvolvimento integrado Visual Studio 2010 Express Edition torna as tarefas orientadas à plataforma Windows mais fáceis.

A interface desenvolvida para desta aplicação é similar a um instalador comum de um programa, onde existem uma série de passos para chegar a uma saída final. Cada passo é uma personalização para o funcionamento do sistema como por exemplo a introdução de variáveis facilmente lidas de uma folha de cálculo como o nome da folha e o nome de uma coluna ou linha em especial.

Um aspecto importante que foi tido em conta na implementação da interface foi a sua consistência. Para conseguir esta consistência o utilizador sabe sempre em que estágio do sistema está com um indicador no topo direito da aplicação. Os botões do programa encontram-se de acordo com os botões do Windows, isto é, os botões "Confirmar", "OK" e "Sim" sempre do lado esquerdo dos botões "Cancelar" e "Não". Ainda existem botões que não se movem de janela para janela ("Anterior" e "Seguinte") que se situam relativamente um ao outro à esquerda e à direita, respectivamente. Um exemplo desta consistência foi também aplicado a áreas de selecção, mais propriamente na escolha das freguesias a pesquisar. Obtêm-se consistência ao convergir a escolha do utilizador à sua selecção, ou seja, começa-se por escolher o distrito, depois o concelho e por fim as freguesias.

Para demonstrar a capacidade de automação do sistema, foi adicionada uma funcionalidade para abrir uma página de testes da Wikipédia com o texto que o utilizador inseriu. Esta funcionalidade serve para pré-visualizar o texto com o código respectivo à Wikipédia antes de executar e organizar os dados do sistema. O programa introduz o texto com códigos da Wikipédia para negrito ou títulos e ainda com as variáveis de dados ([var:nome_da_variavel]) antes de substituir os seus valores. Assim o utilizador pode examinar previamente o texto a ser gerado na procura por erros. Como é mostrado na Figura 17, depois de ter carregado a página *web* requisitada no *WebBrowser* disponível no Visual C#, neste caso uma página de testes da Wikipédia, o texto do utilizador vai ser enviado para a página pelo *internet bot* que irá carregar automaticamente tanto o texto como todas as acções posteriores que um humano tinha de executar como, por exemplo, o clique de um botão.

```
string texto_introduzir = texto;

HtmlDocument doc = this.webBrowser1.Document;
HtmlElement textArea = webBrowser1.Document.All["wpTextbox1"];
string texto_antigo = "<!--não apague esta linha-->{{página de testes}}<!--não
apague esta linha-->\r\n<!--Escreva abaixo da linha! ------------------------------
-- -->\r\n";
string texto_novo = texto_antigo + texto_introduzir;
if (textArea != null) {//adicionar texto do utilizador
    textArea.InnerText = texto_novo;
}
HtmlElement button_preview = webBrowser1.Document.All["wpPreview"]; //nome do botão

if (carregou == 0){//clicar no botão wpPreview
    button_preview.InvokeMember("click");
    carregou = 1;
}
carregou++;
if (carregou >= 2){
    webBrowser1.Show();
}
```

Figura 17- Fragmento de código do *click* de um botão numa página *web*

Na Figura 18 apresenta-se o método para aceder às bases de dados *offline*, mais propriamente a ficheiros de folhas de cálculo em XLS do Excel, realizado através de OLEDB, facilitando assim a codificação para esse acesso de modo que só é necessário saber onde está o ficheiro, o nome da folha, os códigos das freguesias a pesquisar e as colunas onde procurar. A conexão é o fragmento de código mais importante pois define o ficheiro que está a ser acedido e portanto o bom funcionamento do programa depende muito da sua correcta configuração.

```
String connection = String.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0};
Extended Properties=\"Excel 8.0; HDR=YES; IMEX=1; Importmixedtypes=text;
typeguessrows=0;\"", caminho); //configurar a conexão ao ficheiro

OleDbConnection con = new OleDbConnection(connection);
DataTable dt = new DataTable();
OleDbDataAdapter command = new OleDbDataAdapter("select * from [" + folha + "$]", con);
command.Fill(dt);

string[] colunas = procuracolunas.Split(',');

List<string> resultado = new List<string>();

for (int i = 0; i <= dt.Rows.Count - 1; i++){
    if (dt.Rows[i][colunacodigo].ToString() == codigo){
        for (int j = 0; j <= colunas.Length - 1; j++){
            resultado.Add(dt.Rows[i][Convert.ToInt32(colunas[j])].ToString());
        }
    }
}
return resultado;
```

Figura 18 - Código de recolha de bases de dados *offline*

Quanto ao acesso às bases de dados *online*, é executado praticamente todo da mesma forma porque se trata de um acesso a um website específico. Por isso, o acesso é primeiramente composto pelo acesso e recolha da página HTML onde se encontra a informação para posteriormente poder ser tratada com uma simples técnica de HTML *Parsing* como mostra a Figura 19. Esta técnica consiste em estudar onde se encontram os dados e após essa posição ser certificada como verdadeira para todas as outras freguesias naquela página *web*, retira-se toda a informação que não interessa através da procura por elementos chave no código HTML, ou seja, parte-se de um grande texto e vai-se estreitando cada vez o texto mais até se ficar só com a informação desejada.

```
System.Net.WebClient client = new System.Net.WebClient();
client.Encoding = Encoding.UTF8; //Codificação dos caracteres do texto html
string txthtml = "";

try{
    txthtml = client.DownloadString("http://www.anafre.pt/distritos/" + distrito +
"/" + concelho + "/" + freguesia);
}
catch (Exception) {
    //throw;
}

string morada = "";
int indice_morada = txthtml.IndexOf("Morada"); //Descobre o índice da Morada

if (indice_morada != -1) {
    int posdl = txthtml.IndexOf("</dl>", indice_morada);
    morada = txthtml.Substring(indice_morada, posdl - indice_morada);

    int posspan = morada.IndexOf("<span>");
    morada = morada.Substring(posspan);

    Regex r1 = new Regex(@"\s+");
    Regex r2 = new Regex(" ");
    morada = StripTags(morada); //Remove as Tags html
    morada = r2.Replace(morada, @" ");//Remove  
    morada = r1.Replace(morada, @" "); //Remove espaços a mais
    return morada;
}
else{
    return "";
}
```

Figura 19- Fragmento de código de acesso, recolha e análise da informação relativa à morada de uma freguesia

3.4 Testes

Na fase de testes, dois tipos de testes foram aplicados: testes unitários e testes de sistema. Os testes unitários validam entidades simples como por exemplo funções individuais, enquanto os testes de sistema são responsáveis por testar se a aplicação se comporta como foi planeado e definido nos requisitos do sistema (Huizinga & Kolawa, 2007).

A fase de testes unitários envolveu uma plataforma de testes automatizados e alguns testes manuais. Para a fase automatizada de testes o ambiente automatizado usado foi o NUnit versão 2.6.2, com o objectivo de validar que as funções individuais produzem os resultados esperados e não apresentam defeitos ou problemas. Estes testes começaram por analisar as funções e métodos que acediam as bases de dados. Após ter estes dados é possível organizar um simples teste manual para examinar uma saída como, por exemplo, à escolha de uma freguesia conferir se o resultado de determinado método apresenta o mesmo valor daquele guardado na base de dados. Depois disto, os dados gerados numa folha de cálculo são preparados para receber estas saídas, sendo feitos testes similares aos anteriores para assegurar um contínuo bom funcionamento. Um bom exemplo de um destes testes é o teste do método de converter o nome das freguesias de modo a funcionar num endereço, isto é, sem acentuação, espaçamentos ou caracteres especiais. Como se pode ver no código da Figura 20, é usada uma sintaxe específica dos testes e para o teste em si é usada a acentuação mais comum da língua portuguesa, sendo assim codificados de forma a provar o correcto funcionamento e posteriormente são testados na ferramenta NUnit.

```
[TestFixture]
    public class TestClass
    {

        [Test]
        public void teste_completo()
        {
            string texto_nao_tratado = "áàãâ éê í ôõó ú ç";
            string texto_tratado = "aaaa-ee-i-ooo-u-c";

            Assert.AreEqual(texto_tratado, convertelocal(texto_nao_tratado));
        }
```

Figura 20 - Exemplo de código de testes

Como a ferramenta NUnit é automatizada podem-se executar vários testes seguidos sem interrupções. Na Figura 21 são visíveis os testes feitos ao método referido anteriormente.
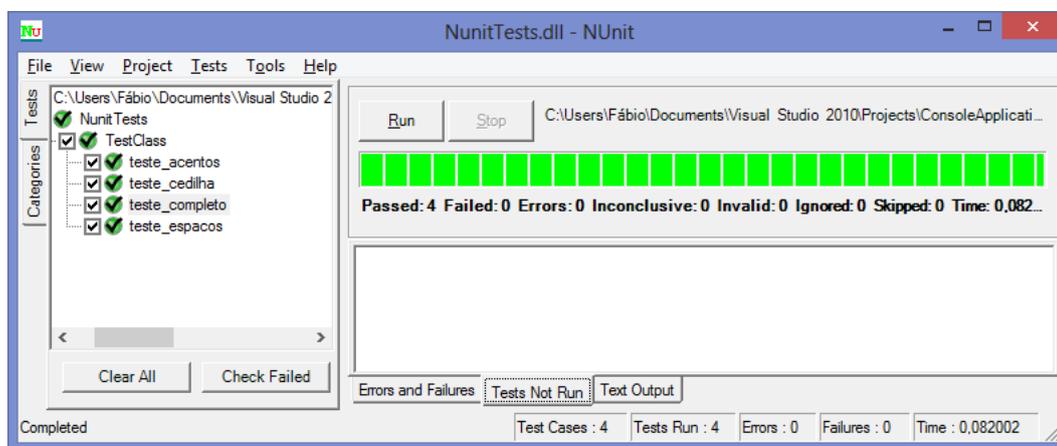
Figura 21- Vários testes a um método com o NUnit

Para testar a escolha de freguesias, o primeiro teste foi executado para uma só freguesia, passando depois a duas e mais freguesias. Esta mesma técnica acompanhou a escolha de concelhos e distritos. Como um concelho é composto por uma ou mais freguesias, parte do código já estava testado, levando assim a um rápido progresso. Uma atenção redobrada foi empregada nas funções que acedem a dados *online*, como as análises HTML de páginas da internet. Esta especial atenção tem a ver com uma vasta gama de problemas relacionados com a ligação aos *sites*, o que pode levar a um mau funcionamento do sistema se não for estudado correctamente.

Os testes de sistema, como o próprio nome indica, envolveram a criação de uma versão do programa de forma a poder realizar estes testes. Os testes de sistema abrangem todas as características da aplicação, mais especificamente, todas as opções de todos os passos do sistema foram usados, simulando assim todos os casos e eventos possíveis. Além de usar dados anormais de entrada para provar que as operações mais comuns não causam um *crash* na aplicação, um outro tipo de teste aplicado é conhecido por testes de lançamento. Este tipo de teste tem como objectivo assegurar que o sistema faz o que é suposto fazer de acordo com os requisitos mas mais focado numa aproximação aos chamados testes de cenário. Nestes testes vários cenários de utilização foram concebidos, assim como foram desenvolvidos os seus casos de teste. Por exemplo, ao selecionar uma (ou algumas) freguesia e guardando os resultados para um único (ou vários) ficheiro, foram cenários ou caminhos em que o sistema é normalmente usado.

3.5 Tutorial da Aplicação

Como em qualquer outro programa, é necessário criar um manual de utilização ou tutorial de modo a dar a conhecer ao utilizador as capacidades da ferramenta.

A interface deste programa foi desenhada a pensar no utilizador devido a ser bastante simples e intuitiva. Pelo lado simples, usa ficheiros ".xls" como bases de dados que são bastante comuns quando se usam tabelas e folhas de cálculo. É intuitivo devido ao facto de se utilizar uma série de passos para se chegar ao conteúdo final como se fosse um simples instalador de programas. Posto isto, foram criados cinco passos para chegar a um resultado completo.

No primeiro passo, a aplicação necessita que lhe seja introduzido um caminho de um ficheiro ".xls" para que todo o programa funcione (ponto 1 da Figura 22). Isto acontece deste modo pois a chave primária que interliga todas as tabelas tem por base um único ficheiro. Assim é possível que no passo para selecionar a localidade se possa ter um ficheiro sempre actualizado, pois é o mesmo deste passo. Para aceder ao ficheiro, especifica-se o nome da folha à qual se quer aceder, pois um ficheiro pode ter várias folhas (ponto 2 da Figura 22). A tabela em si deve conter no mínimo uma coluna com os distritos, uma com os concelhos, uma com as freguesias e uma outra com a coluna chave que é por sua vez a coluna com os códigos das freguesias (ponto 3 da Figura 22). Com a especificação destas variáveis de entrada, tendo em conta que a coluna número zero corresponde à primeira coluna (coluna "A"), pode-se atribuir qualquer ficheiro desde que cumpra estes requisitos mínimos (ponto 3 da Figura 22). Por outro lado, o número da linha de começo dos códigos é o número correspondente na folha de cálculo (ponto 4 da Figura 22). Para complementar este passo, após o utilizador avançar para o próximo passo estes dados são guardados automaticamente num ficheiro de texto e com as variáveis separadas por ponto e vírgula, para que da próxima vez que o programa seja usado já esteja configurado.
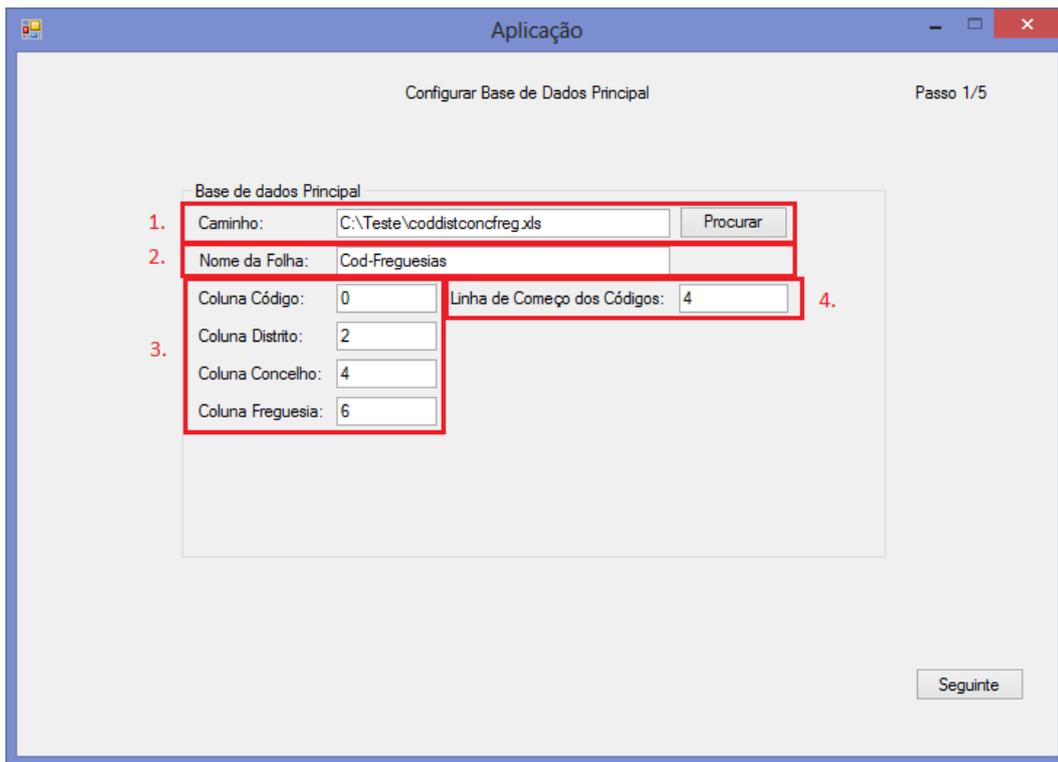
Figura 22- Primeiro passo da aplicação

Na Figura 23 percebe-se que no segundo passo o utilizador tem à disposição uma interface para gerir bases de dados na aplicação, isto é, assim é possível adicionar mais informação com objectivo de que esta seja a mais actualizada e completa possível (ponto 1 da Figura 23). Com a adição de novas bases de dados é possível seleccionar quais as colunas que vão ser procuradas por informação, mas para isso requer o número da coluna dos códigos das freguesias (ponto 2 da Figura 23). A especificação das tabelas inerentes aos ficheiros é muito importante para o correcto funcionamento do programa e por isso as colunas devem ser nomeadas. A atribuição de nome às colunas do ficheiro é extremamente simples pois cada linha da área de texto corresponde ao nome de uma coluna (ponto 3 da Figura 23). Neste ponto é preciso ter algum cuidado pois por vezes podem existir colunas ocultas e a sua não inserção causa uma saída de dados da aplicação inesperada. Após a atribuição é possível fazer a selecção das colunas que se pretendem examinar no resultado final (ponto 4 da Figura 23).
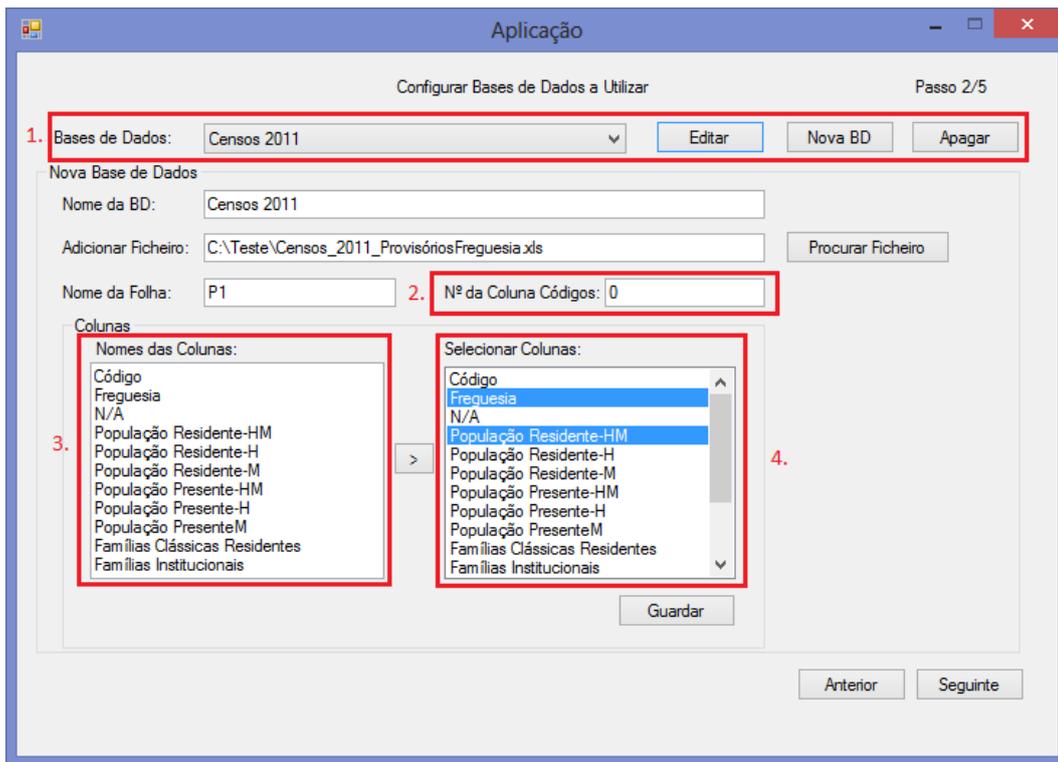
Figura 23 - Segundo passo da aplicação

As especificações que o utilizador insere nesta interface de gestão da base de dados são guardadas de uma forma similar ao primeiro passo, ou seja, os dados são separados por ponto e vírgula e guardados em formato texto, o que faz com que este processo trabalhe numa forma mais eficaz para que quando os dados forem necessários sejam chamados de forma mais simples. A gestão das bases de dados tem por base o seu nome e quando estas têm o mesmo nome, os novos dados são sobrepostos aos antigos, eliminando assim qualquer conflito e ambiguidade entre elas. É possivel inserir bases de dados com o mesmo ficheiro, desde que os nomes sejam diferentes, o que pode acontecer quando num ficheiro existam várias folhas.

No terceiro passo e de acordo com o ficheiro introduzido no passo um, vai surgir uma área para seleccionar os distritos, concelhos e freguesias tendo em conta o previamente seleccionado, ou seja, ao escolher um distrito surgem na lista ao lado os concelhos associados a esse distrito e o mesmo acontece entre os concelhos e freguesias. Com este método é muito rápida a selecção da localidade porque vai-se cada vez mais especificando o que se ambiciona, como se pode ver na Figura 24, sendo a sua selecção visualmente intuitiva. Mais concretamente, ao selecionar um distrito, vão surgir todos os concelhos desse distrito na coluna dos concelhos (ponto 2 da Figura 24), mas ao ser

escolhido mais que um distrito impossibilita a escolha de concelhos dos mesmos. No caso de escolher apenas um distrito e nenhum concelho, a procura resume-se a esse mesmo distrito fazendo uma pesquisas de todas as freguesias do distrito (ponto 1 da Figura 24).
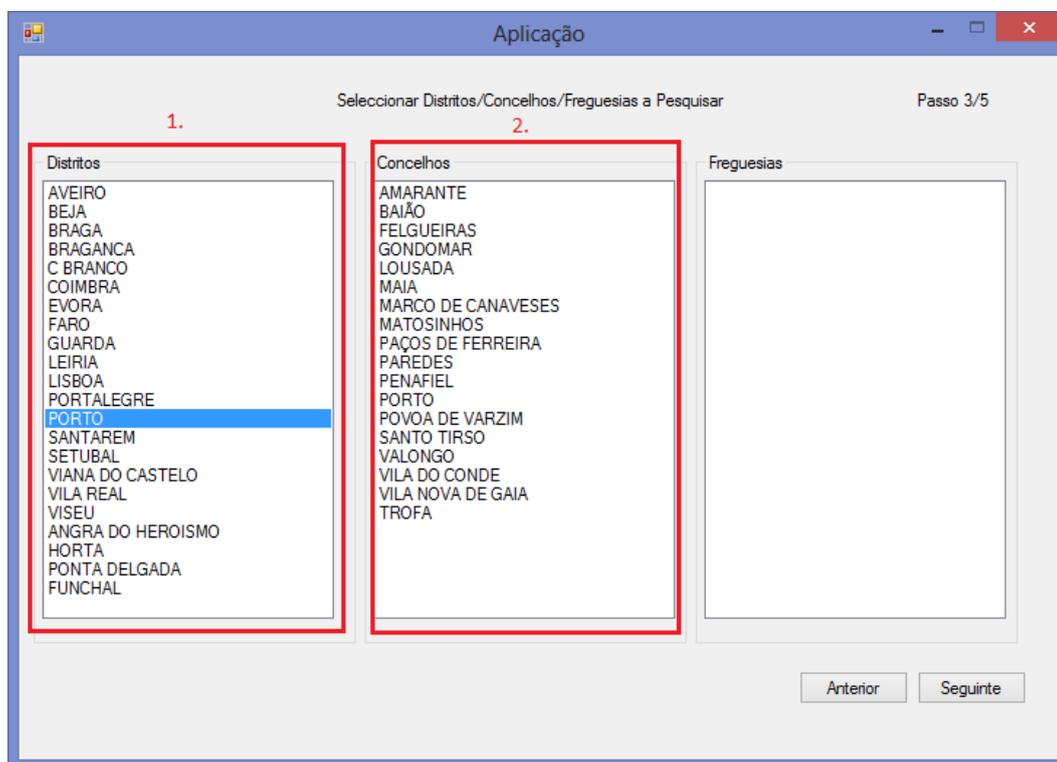
Figura 24- Terceiro passo da aplicação (Distritos/Concelhos)

O mesmo aplica-se de forma similar na fase seguinte à selecção do distrito, ao selecionar um concelho vão aparecer todas as freguesias desse concelho na coluna das freguesias (ponto 2 da Figura 25), mas se a selecção incluir mais que um concelho deixa de ser possível, como no caso dos distritos, de escolher as freguesias associadas aos vários concelhos. Como no caso da coluna dos distritos, para a coluna dos concelhos acontece o processo semelhante de selecção, isto é, quando se selecciona apenas um concelho, são pesquisadas todas as freguesias desse concelho (ponto 1 da Figura 25). Para o caso da coluna das freguesias é diferente pois como são as últimas escolhas, podem ser seleccionadas várias porque não têm uma cadeia subsequente que possam afectar.
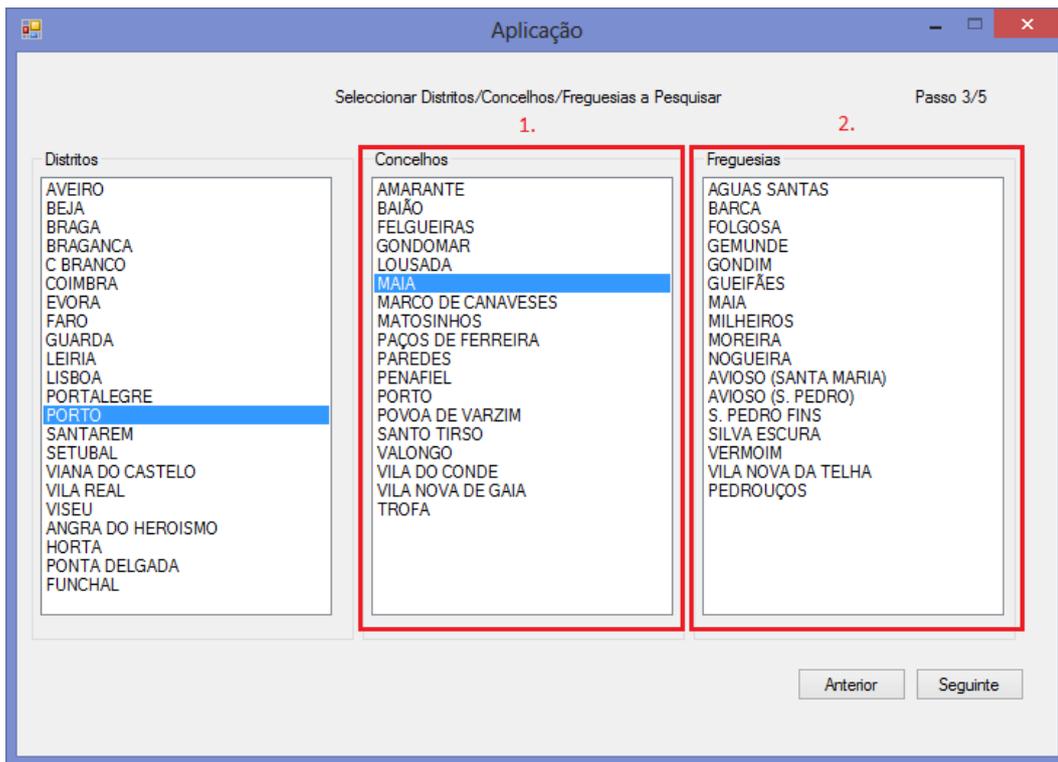
Figura 25- Terceiro passo da aplicação (Concelhos/Freguesias)

No quarto passo são definidas as bases de dados a usar, pois estas podem ter sido adicionadas mas o seu uso não é requerido e também é definido se vai ser necessário guardar em texto o resultado final da aplicação. Ao escolher guardar em texto, é disposta a escolha de ser tudo escrito num único ficheiro, por freguesia, por concelho ou por distrito para facilitar a procura do texto ao utilizador (ponto 2 da Figura 26). Quanto à escolha das bases usados a usar, fundamenta-se nas bases de dados introduzidas e guardadas no segundo passo (ponto 1 da Figura 26). Para complementar as bases de dados em ficheiros, ou seja as bases de dados *offline*, existe um conjunto de bases de dados *online* opcionais que podem ser seleccionadas. Entre estas bases de dados encontram-se tabelas de gentílicos da Wikipédia, contactos, tais como telefone, *fax* e *e-mail* e ainda a morada e página da internet (ponto 3 da Figura 26). Os contactos, morada e página web provêm de uma base de dados da Associação Nacional de Freguesias (ANAFRE).
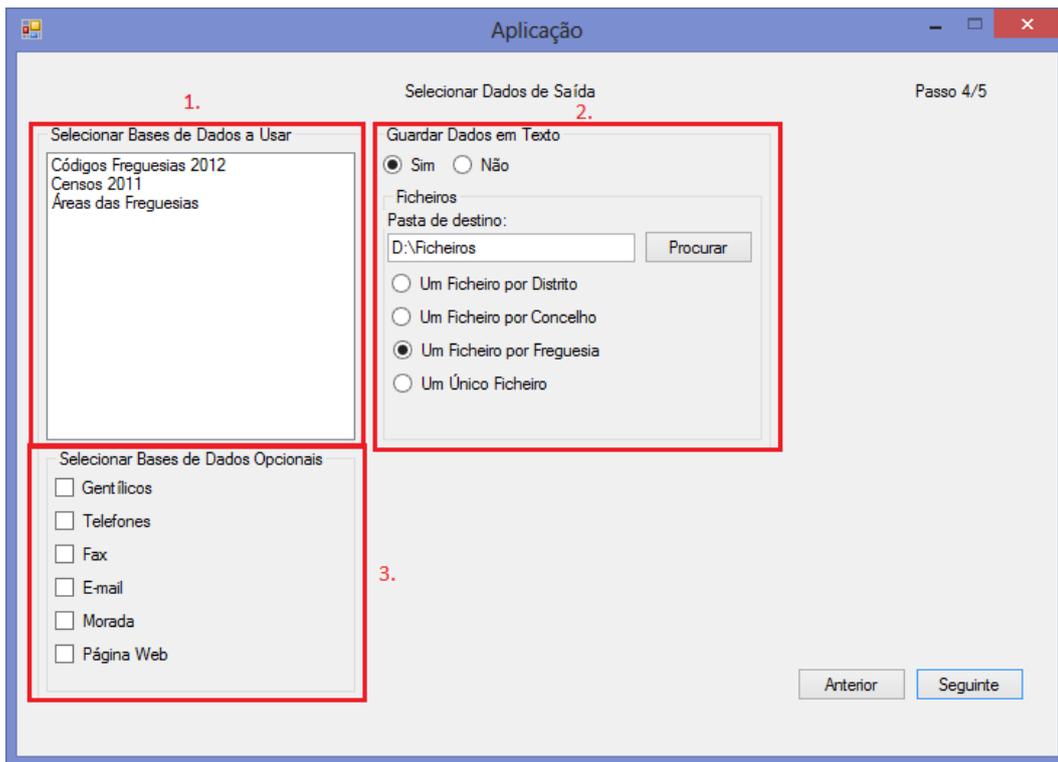
Figura 26 - Quarto passo da aplicação

No quinto passo, se for escolhido guardar em texto, é editado o texto de saída para todas as freguesias usadas (ponto 3 da Figura 27). Nesta edição está incluído um sistema que permite introduzir variáveis no texto que posteriormente são substituídas com os valores para cada freguesia (ponto 2 da Figura 27). Neste passo também é usado um sistema de gestão semelhante ao usado no segundo passo para gerir os diferentes textos que o utilizador possa ter.
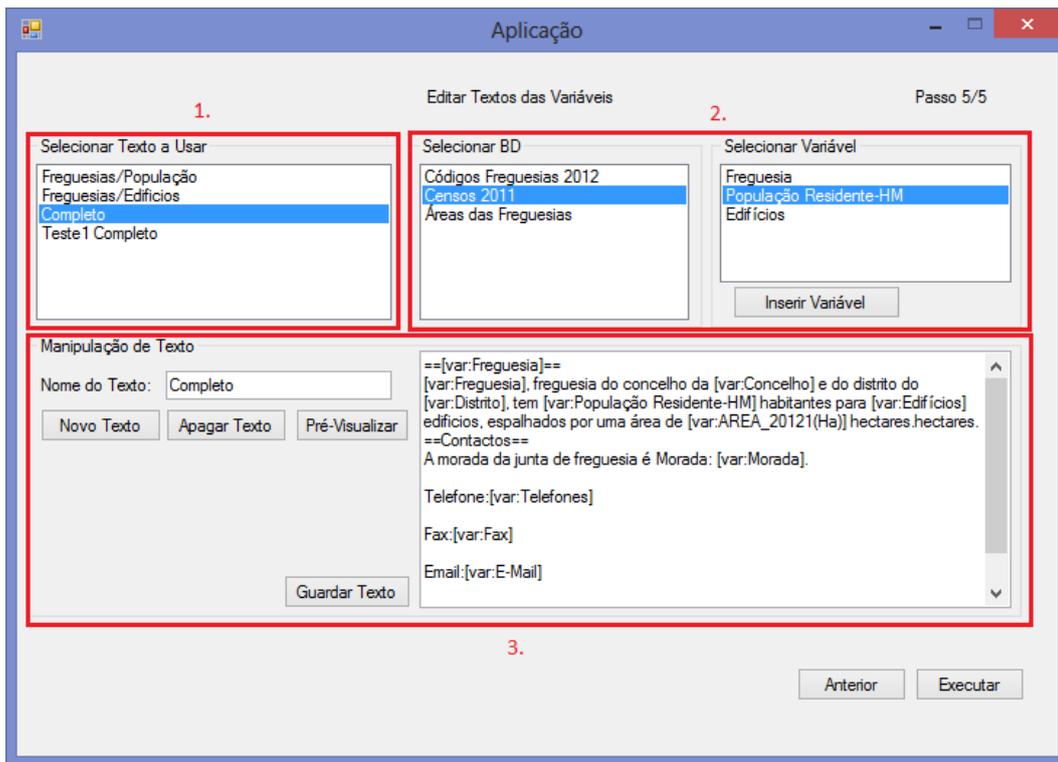
Figura 27 - Quinto passo da aplicação

Ao editar o texto é possível fazer uma pré-visualização numa página de testes da Wikipédia. O que acontece é a abertura de um navegador (*browser*) que introduz e mostra na Wikipédia o texto depois de compilado (Figura 28). Esta funcionalidade mostra a facilidade de aceder a conteúdos *online* de maneira a editar automaticamente uma página da Wikipédia.

Figura 28- Pré-visualização do texto na Wikipédia

No sexto e último passo faz-se a apresentação de resultados numa tabela e, caso a opção “Guardar Dados em texto” seja “Sim”, encontra-se um botão de atalho para a pasta de destino dos mesmos, como se pode ver o resultado na Figura 29. Como a tabela é a saída mais importante do programa, pois é a partir daí que todos os dados são retirados, é necessário ter todos os cabeçalhos de coluna bem colocados para não existirem trocas de informação. Para isso, os cabeçalhos ficam com os nomes das colunas que o utilizador escreveu e seleccionou no segundo passo (ponto 1 da Figura 29). Quanto aos dados associados a cada freguesia, provêm da busca das bases de dados *offline* de acordo com o código de freguesia e *online* de acordo com o nome da freguesia e posicionamento do texto. A saída de texto é escrita como o utilizador redigiu no quinto passo, mas aparecem para todas as freguesias e com as variáveis já substituídas pelas das bases de dados. Como o processo de recolha de dados pode ser demorado, existe uma barra de progresso onde se pode ver o estado da aplicação. Quando a aplicação passa para bases de dado *online* aparece também o nome da freguesia que está a ser pesquisada para tornar mais apelativa a espera (ponto 2 da Figura 29).

Código	Distrito	Concelho	Freguesia	População Residente-HM	Edifícios	AREA_20121(Ha)
130601	PORTO	MAIA	Águas Santas	27470	4495	822,59
130602	PORTO	MAIA	Barca	2633	834	329,31
130603	PORTO	MAIA	Folgosa	3704	1313	1011,45
130604	PORTO	MAIA	Gemunde	5215	1131	545,06
130605	PORTO	MAIA	Gondim	2208	623	139,24
130606	PORTO	MAIA	Gueifães	11964	2500	286,56
130607	PORTO	MAIA	Maia	12406	1396	342,57
130608	PORTO	MAIA	Milheirós	4861	1263	360,42
130609	PORTO	MAIA	Moreira	12890	2459	867,27
130610	PORTO	MAIA	Nogueira	5473	1243	350,42
130611	PORTO	MAIA	Avioso (Santa M...	4513	787	488,11
130612	PORTO	MAIA	Avioso (São Pedro)	3826	823	489,32
130613	PORTO	MAIA	São Pedro Fins	1837	555	470,68
130614	PORTO	MAIA	Silva Escura	2507	720	532,24

Figura 29 - Visualização dos resultados em tabela

Uma das maneiras possíveis de guardar os dados em ficheiros é visível na Figura 30, em que se pode facilmente procurar os ficheiros pelo nome das freguesias e aceder ao conteúdo em texto já tratado pela aplicação. Para além disso têm-se acesso à última tabela que o programa construiu, podendo ser reutilizada no programa com uma simples conversão.

Figura 30 - Ficheiros de saída da aplicação

4. Conclusões

Desenvolveu-se um *software* automático de recolha de informação de várias fontes *online* e *offline* sobre freguesias portuguesas. Este desenvolvimento passa pelas quatro actividades fundamentais de engenharia de *software*: especificação de requisitos, desenho, implementação e testes. Dada a natureza do trabalho foi possível submeter e apresentar um artigo sobre o mesmo numa conferência internacional (Oliveira & Pereira, 2013).

O programa usa uma base de dados principal com uma lista de todos os distritos, concelhos e freguesias de modo a estabelecer uma relação com todas as outras bases de dados, permitindo a extracção de informação. O utilizador pode selecionar uma combinação de um distrito, um ou mais concelhos de um distrito, ou uma ou mais freguesias de um concelho. Para além do utilizador poder selecionar as áreas geográficas para pesquisa, também pode controlar as características da informação a reunir, incluindo assim os dados relativos à população, tais como: faixas etárias, estado civil, edifícios ou composição do agregado familiar. Duas das principais bases de dados usadas contêm estatísticas do Censos 2011 elaborada pelo Instituto Nacional de Estatística e dados geográficos do Instituto Geográfico Português.

Esta aplicação pode ser usada, por exemplo, por empresas de marketing que necessitem de comparar dados específicos entre freguesias. Pode também ser usada por editores da Wikipédia que pretendam actualizar os dados dos artigos sobre as freguesias portuguesas, bem assim como por qualquer pessoa que precisem de aceder a dados estatísticos nacionais. Como esta ferramenta possui uma componente *online* opcional, outros países com o mesmo modelo administrativo de território podem usá-la, mas este pormenor não foi confirmado.

Como trabalho futuro, o uso de um “*internet bot*” mais geral poderia ser desenvolvido de modo a que a aplicação não seja tão específica e com vista a poder incluir uma maior variedade de bases de dados. Pode também ser implementada uma habilidade para copiar as bases de dados *online* para a máquina local e assim poder acedê-las futuramente no modo *offline*. O desenvolvimento de um método para actualizar automaticamente a informação das freguesias portuguesas na Wikipédia também seria de considerar. Por fim, podia ser conduzido um estudo com empresas de modo a melhorar a aplicabilidade da ferramenta.

O resultado deste trabalho específico para um caso em particular, demonstra a construção de uma ferramenta que qualquer pessoa pode utilizar para recolher informações sobre freguesias portuguesas.

5. Referências

Ahn, L. v., Blum, M., Hopper, N. J., & Langford, J. (2003). *"CAPTCHA: Using Hard AI Problems for Security"*. Obtido de http://www.captcha.net/captcha_crypt.pdf

AR. (30 de Maio de 2012). Lei n.º 22/2012 de 30 de Maio - Aprova o regime jurídico da reorganização administrativa. In *Diário da República* (pp. 2826-2836).

Arian, A. (s.d.). *Internet Communication, Protocols and related subjects.* Obtido de http://books.google.pt/books?id=J1gb2eb-NuEC

ATA. (2010). Obtido de Portal das Finanças: http://info.portaldasfinancas.gov.pt/NR/rdonlyres/B365CA49-D07B-4FCD-B9BF-8FADCA0EF528/0/coddistconcfreg.xls

Basili, V. R. (1990). *Viewing maintenance as reuse-oriented software development*. *IEEE Computer Society Press*, 19-25.

Brin, S., & Page, L. (1997). *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Obtido de http://infolab.stanford.edu/~backrub/google.html

Carter, C. (2010). *User:ClueBot NG*. Obtido de http://en.wikipedia.org/wiki/User:ClueBot_NG

Castillo, C. (2004). *Effective Web Crawling.*

Google Inc. (2013). *What is reCAPTCHA.* Obtido de http://www.google.com/recaptcha/learnmore

Gray, M. (1996). *Credits and Background*. Obtido de *Internet growth and statistics*: http://www.mit.edu/people/mkgray/net/background.html

Heaton, J. (2002). *Programming Spiders, Bots, and Agregators in Java.* ISBN-13: 978-0782140408. Sybex.

Huizinga, D., & Kolawa, A. (2007). *Automated Defect Prevention: Best Practices in Software Management.* Wiley-IEEE Computer Society Pr.

IGP. (2011). *Carta Administrativa Oficial de Portugal.* Obtido de Instituto Geográfico Português: http://www.igeo.pt/produtos/cadastro/caop/download/Dados_CAOP2011.pdf

IGP. (2012). *Área das Freguesias, Municípios e Distritos da CAOP2012.* Obtido de Instituto Geográfico Português: http://www.igeo.pt/produtos/cadastro/caop/download/Areas_Freg_Mun_Dist_CAOP20121.zip

INE. (2012). *Censos 2011*. Obtido de Instituto Nacional de Estatística: http://censos.ine.pt/

Knight, W. (2005). *Software bots could menace Google ads.* Obtido de NewScientist: http://www.newscientist.com/article/dn6972-software-bots-could-menace-google-ads.html#.UdbgOKxnCgE

Ledford, J. (2008). *Google Adsense for Dummies.* ISBN-13: 978-0470292891. Wiley Publishing, Inc.

Lista de Freguesias de Portugal. (2013). Obtido de http://pt.wikipedia.org/wiki/Anexo:Lista_de_freguesias_de_Portugal

Lista de Gentílicos de Portugal. (2013). Obtido de http://pt.wikipedia.org/wiki/Anexo:Lista_de_gent%C3%ADlicos_de_Portugal

Nasaw, D. (2012). *Meet the 'bots' that edit Wikipedia.* Obtido de BBC News Magazine: http://www.bbc.co.uk/news/magazine-18892510

Oliveira, F., & Pereira, V. (2013). *Development of an automated tool to consolidate information about Portuguese administrative parishes. 3rd International Conference on Managing Services in the Knowledge Economy 2013*.

Olston, C., & Najork, M. (2010). *Web Crawling: Foundations and Trends in Information Retrieval.* Now Publishers Inc.

Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach, Seventh Edition.* ISBN-13: 978-0073375977. McGraw-Hill.

Puri, R. (2003). *Bots &; Botnet: An Overview.* Obtido de http://www.sans.org/reading_room/whitepapers/malicious/bots-botnet-overview_1299

Ramsey, D. (2010). *User: rambot*. Obtido de http://en.wikipedia.org/wiki/User:Rambot

Silva, A. C. (16 de Janeiro de 2013). *Mensagem do Presidente da República à Assembleia da República sobre a Reorganização Administrativa do Território das Freguesias.* Obtido de Página Oficial da Presidência da República Portuguesa: http://www.presidencia.pt/?idc=10&idi=71113

Smith, P. R., & Sarfaty, R. (1993). *Creating a strategic plan for configuration management using computer aided software engineering (case) tools*. *DOE Facilities 2000: Planing for Change*.

Sommerville, I. (2010). *Software Engineering Ninth Edition.* ISBN-13: 978-0137035151. Addison-Wesley.

Weys, C. (2011). *User:Cydebot*. Obtido de http://en.wikipedia.org/wiki/User:Cydebot

Wikipedia bots. (2012). Obtido de http://en.wikipedia.org/wiki/Category:Wikipedia_bots

Wikipedia bots by status. (2012). Obtido de http://en.wikipedia.org/wiki/Category:Wikipedia_bots_by_status

Wikipedia: Bots. (18 de April de 2013). Obtido de Wikipedia: http://en.wikipedia.org/wiki/Wikipedia:Bots

Wikipedia: Bot policy. (2012). Obtido de http://en.wikipedia.org/wiki/Wikipedia:Bot_policy

6. Anexo 1

Método de preencher as listas de escolha das zonas a pesquisar

```
public static void preenche_listas(string caminho, string folha, int
coluna_local_maior, int coluna_local_menor, ListBox list_item_procurar, ListBox
list_preencher){
        //local_maior é por exemplo um distrito e a zona menor é a zona imediatamente
        inferior,neste caso um concelho
        //funciona também para freguesias seguindo a mesma regra, o local maior passa
        a concelho e o menor a freguesia
        //a coluna_local_maior é o número da coluna da folha de cálculo onde se
        encontra a coluna do local maior

        String connection = String.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data
        Source={0}; Extended Properties=\"Excel 8.0; HDR=YES; IMEX=1;
        Importmixedtypes=text; typeguessrows=0;\"", caminho);
        OleDbConnection con = new OleDbConnection(connection);

        DataTable dt = new DataTable();
        OleDbDataAdapter command = new OleDbDataAdapter("select * from [" + folha +
        "$]", con);

        command.Fill(dt);
        //zona_procura é o que o utilizador selecionou na lista
        string zona_procura = list_item_procurar.SelectedItem.ToString();

        for (int i = 0; i <= dt.Rows.Count - 1; i++){
                if (dt.Rows[i][coluna_local_maior].ToString() == zona_procura){
                    if
        (!list_preencher.Items.Contains(dt.Rows[i][coluna_local_menor])){
                        list_preencher.Items.Add(dt.Rows[i][coluna_local_menor]);
                    }
                }
        }
}
```